D2.4 Performance of WHERE Cooperative Positioning Techniques

Nielsen, Jimmy Jessen; Arambasic; Bastos; Denis; Hadzic; He; Laaraiedh; Ma; Maman; Mensing; Pedersen, Claus; Rodriguez; Savic; Uguen; Zazo

Publication date:
2010

Document Version
Early version, also known as pre-print

ICT-217033 WHERE

D2.4 Version 1.0

Performance of WHERE Cooperative Positioning Techniques

Contractual Date of Delivery to the CEC:	*30/04/2010*
Actual Date of Delivery to the CEC:	**10/05/2010**
Editor:	**Yi Ma**
Author(s):	*Igor Arambasic, Joaquim Bastos, Benoît Denis, Senka Hadzic, Ziming He, Mohamed Laaraiedh, Yi Ma, Mickael Maman, Christian Mensing, Jimmy Jessen Nielsen, Claus Pedersen, Jonathan Rodriguez, Vladimir Savic, Bernard Uguen, Santiago Zazo*
Participant(s):	*AAU, CEA, DLR, UR1, IT, UPM, UoS*
Work package:	**WP2 – Hybrid and Cooperative Positioning**
Security:	**PU**
Nature:	**R**
Version:	1.0
Total number of pages:	123

Abstract: This deliverable presents results about novel cooperative positioning schemes and performance analysis from the communication aspects. The focus is on small-scale indoor scenarios with ad hoc sensor network and impulse radio (IR) ultra wideband (UWB), mid-scale indoor scenarios with WiFi systems, and large-scale outdoor scenarios such as cellular communication systems. It is shown that the proposed cooperative positioning schemes such as iterative cooperative positioning algorithms, generalized belief propagation, mean field algorithm, and multidimensional scaling can significantly improve the accuracy of positioning information in the small-scale scenarios, at the cost of communication overhead. Moreover, the benefit of cooperation is verified using IR-UWB hardware platforms. The overhead/performance trade-off is carefully investigated for cooperative positioning in mid-scale indoor and large-scale outdoor scenarios. Finally, medium access schemes for cooperative positioning are proposed to reduce the communication overhead and latency but maintain the overall accuracy of location information.

Keyword list: Cooperative positioning, hybrid data fusion, tracking, UWB, WiFi, cellular, GNSS, TOA, ranging, CRLB, LOS, belief propagation

Disclaimer:

Executive Summary

One of the primary objectives of the WHERE project is to study radio positioning techniques using existing and future heterogeneous communication networks. In order to offer reliable and accurate location information, the WHERE consortium researches novel cooperative positioning schemes as well as the fundamental performance limits. Specifically, this work includes the following three main tasks:

1) To investigate efficient and reliable cooperative positioning techniques for various communication scenarios covering small-scale and mid-scale indoor systems and large-scale outdoor systems.
2) To develop appropriate and meaningful analytical models for the performance analysis of cooperative positioning schemes in heterogeneous communication networks.
3) To investigate overhead, latency and performance trade-off of cooperative positioning schemes from the communication point-of-view.

In general, the presented results about positioning accuracy and communication overhead should serve as an input to WP3. The major contributions of this work are in two folds: novel cooperative positioning schemes for various communication architectures and performance analysis from the communication aspects. The focus of this deliverable is on four WHERE scenarios addressed in Section 1. It is shown that the proposed cooperative positioning schemes such as iterative cooperative positioning algorithms, generalized belief propagation (GBP), mean field algorithm, and multidimensional scaling can significantly improve the accuracy of positioning information in the small-scale scenarios, respectively, at the cost of communication overhead. Besides, the actual benefit of cooperation is also verified using WHERE WP5 Impulse Radio – Ultra Wideband (IR-UWB) hardware platforms. Trade-off between positioning accuracy and communication overhead is carefully investigated for both the mid-scale (WiFi) and large-scale scenarios (cellular). Finally, new decentralized and prioritized medium access schemes, providing support to cooperative tracking, are proposed to enhance location precision under partial connectivity, while limiting consumed energy and time resources. The performance of the algorithms is evaluated in terms of realistic error models (by taking into account the work performed in WHERE WP2 Task 2.1 and WHERE WP4) and by post-processing of measured data from WHERE WP5.

Specifically, we present the scenarios in Section 1 that are used to investigate the performance of cooperative positioning schemes. The focus in this deliverable is on the small-scale indoor scenarios T1.A and T1.B, the mid-scale indoor scenario T2.A, and the large-scale outdoor scenario T3.

Section 2 presents several novel cooperative positioning schemes for various wireless environments. The focus of Section 2.1, Section 2.2, Section 2.3, Section 2.6 and Section 2.7 is on the scenarios T1.A or T1.B. Specifically, Section 2.1 presents multi-hop cooperative positioning schemes for ad hoc networks. Section 2.2 presents static indoor multi-hop positioning scheme using GBP. In Section 2.3, the mean field algorithm is investigated as an alternative to the nonparamteric belief propagation algorithms. Section 2.6 presents cooperative positioning using multidimensional scaling. Section 2.7 verifies cooperative positioning experimentally based on IR-UWB platforms. In addition, Section 2.4 investigates cooperative positioning and tracking algorithms in WiFi (scenario T2.A). Section 2.5 presents cooperative positioning for cellular radio systems (scenario T3).

Section 3 presents the performance analysis for novel cooperative positioning schemes from the communication aspects. Specifically, Section 3.1 investigates effect of delays in WiFi-based (scenario T2.A) cooperative positioning. Section 3.2 presents new prioritized and decentralized medium access schemes for cooperative tracking based on short-range radio links (scenario T1.A).

Authors

Partner	Name	Phone / Fax / e-mail
AAU		
	Jimmy Jessen Nielsen	Phone: +45 99409867 Fax: N/A e-mail: jjn@es.aau.dk
	Claus Pedersen	Phone: +45 99408615 Fax: N/A e-mail: cpe@es.aau.dk
CEA		
	Benoît Denis	Phone: +33 438781821 Fax: +33 438786586 e-mail: benoit.denis@cea.fr
	Mickael Maman	Phone: +33 438781045 Fax: +33 438786586 e-mail: mickael.maman@cea.fr
DLR		
	Christian Mensing	Phone: +49 8153282878 Fax: +49 8153281871 e-mail: christian.mensing@dlr.de
IT		
	Jonathan Rodriguez	Phone: +351 234377900 Fax: +351 234377901 e-mail: jonathan@av.it.pt
	Joaquim Bastos	Phone: +351 234377900 Fax: +351 234377901 e-mail: jbastos@av.it.pt
	Senka Hadzic	Phone: +351 234377900 Fax: +351 234377901 e-mail: senka@av.it.pt
UoS		
	Ziming He	Phone: +44 1483683601 Fax: +44 1483686011 e-mail: z.he@surrey.ac.uk
	Yi Ma	Phone: +44 1483683609 Fax: +44 1483686011 e-mail: y.ma@surrey.ac.uk
UPM		
	Vladimir Savic	Phone: +34 915495700 Fax: +34 913367350 e-mail: vladimir@gaps.ssr.upm.es
	Santiago Zazo	Phone: +34 915495700 Fax: +34 913367350 e-mail: santiago@gaps.ssr.upm.es
	Igor Arambasic	Phone: +34 915495700 Fax: +34 913367350 e-mail: igor@gaps.ssr.upm.es

Partner	Name	Phone / Fax / e-mail
UR1		
	Mohamed Laaraiedh	Phone: +33 223235075 Fax: +33 223235616 e-mail: mohamed.laaraiedh@univ-rennes1.fr
	Bernard Uguen	Phone : +33 223236033 Fax : +33 223235616 e-mail : bernard.uguen@univ-rennes1.fr

Table of Contents

List of Acronyms and Abbreviations 9

1. Introduction 11

2. Cooperative Positioning Techniques 15

2.1 Cooperative Positioning for Ad Hoc Networks 15
2.1.1 Introduction 15
2.1.2 Localization in Ad Hoc Networks 15
2.1.3 Multi-Hop Localization 15
2.1.4 Iterative Framework for Cooperative Positioning 16
2.1.5 Impact of Anchor Placement 20
2.1.6 Conclusion 22
2.2 Cooperative Positioning using Generalized Belief Propagation Techniques 22
2.2.1 Introduction 22
2.2.2 Overview of Localization using Nonparametric Belief Propagation (NBP) Algorithm 23
2.2.3 Correctness of Belief Propagation 28
2.2.4 Generalized Belief Propagation Based on Kikuchi Approximation (GBP-K) 30
2.2.5 Generalized Belief Propagation Based on Junction-tree Method (GBP-JT) 33
2.2.6 Nonparametric Generalized Belief Propagation Based on Junction-tree Method (NGBP-JT) 35
2.2.7 Nonparametric Belief Propagation Based on Spanning Trees (NBP-ST) 41
2.2.8 RSS/TOA Based Distance Modeling 45
2.2.9 Conclusion and Future Work 53
2.3 A Mean Field Algorithm for Sensor Self-Localization in Cooperative Wireless Networks 54
2.3.1 Model 54
2.3.2 Variational Methods and Mean Field Theory 55
2.3.3 Mean Field Localization Algorithm 56
2.3.4 Numerical Results 57
2.3.5 Conclusion and Future Work 59
2.4 Centralized Cooperative Positioning and Tracking 59
2.4.1 System Model 60
2.4.2 Centralized Cooperative Positioning Algorithms 62
2.4.3 Simulation Results 64
2.4.4 Conclusion 68
2.5 Cooperative Localization in OFDMA-Based Cellular Networks 68
2.5.1 Cooperative Localization Method for Case when the Number of Located MTs is Larger than Three 70
2.5.2 Cooperative Localization Method for Case when the Number of Located MTs is One 77
2.5.3 Conclusion 83
2.6 Cooperative Localization Based on Multidimensional Scaling 84
2.6.1 Introduction 84
2.6.2 Localization using Multidimensional Scaling Technique 84
2.6.3 Simulations and Results 85
2.6.4 Conclusion 89
2.7 Experimental Verification of Cooperation Benefits in Positioning with Biased TOA-Based Range Measurements 90
2.7.1 Measurement Campaign Supporting Off-Line Cooperative Positioning 90
2.7.2 Range Measurements and Tested Positioning Agorithms 91
2.7.3 Off-Line Cooperative Positioning Results 91
2.7.4 Conclusion and Perspectives 93

3. Communication Aspects of Cooperative Positioning 95

3.1 Realistic Communications Constraints for Centralized Cooperative Positioning and Tracking 95
3.1.1 Measurement Collection for Device-Based Conventional Localization 95
3.1.2 Measurement Collection for Centralized Cooperative Localization 95
3.1.3 802.11a WiFi Network Model 97
3.1.4 Group Mobility Model 97

3.1.5 Evaluation Methodology 98
3.1.6 Simulation Results 98
3.1.7 Conclusion 101
3.2 Extension of Prioritized and Decentralized Medium Access Schemes into Cooperative Tracking Scenarios 101
3.2.1 Motivations and Goals 101
3.2.2 Prioritized Medium Access Schemes 102
3.2.3 Decentralized Cooperative Tracking Filter 107
3.2.4 Results 109
3.2.5 Conclusion and Perspectives 116

4. Conclusion and Future Work 117

4.1 Conclusion 117
4.2 Future Work 118

5. References 121

List of Acronyms and Abbreviations

Term	Description
AN	Anchor Node
AOA	Angle Of Arrival
AP	Access Point
AWGN	Additive White Gaussian Noise
BP	Belief Propagation
BRR-BA	Broadcasted Ranging Requests-Broadcasted Answers
BS	Base Station
CDF	Cumulative Distribution Function
CP	Cooperative Positioning
CRLB	Cramer-Rao Lower Bound
DLS	Distributed Least Square
EKF	Extended Kalman Filter
FCEC	Fully Centralized and Exhautively Cooperative medium access
FCSC	Fully Centralized and Selectively Cooperative medium access
FDNC	Fully Decentralized Non Cooperative medium access
GDOP	Geometric Dillution Of Precision
GN	Gauss-Newton
GPS	Global Positioning System
GTS	Guaranteed Time Slots
HCSC	Half-Centralized and Selectively Cooperative medium access
HDF	Hybrid Data Fusion
HS	Hot Spot
IR	Impulse Radio
KF	Kalman Filter
LDC	Low Duty Cycle
LDR	Low Data Rate
LMDS	Localization with Multidimensional Scaling
LOS	Line-of-Sight
LS	Least Squares
LTE	Long Term Evolution
MAC	Medium Access Control
MDS	Multidimensional Scaling
MAP	Maximum A Posteriori
MMSE	Minimum Mean Square Error
MS	Mobile Station
MT	Mobile Terminal
NC	Non Cooperative medium access
NLOS	None Line of Sight
NBP	Non Parametric Belief Propagation
KDE	Kernel Density Estimate
KF	Kalman Filter
P2P	Peer to Peer
PNC	PicoNet Coordinator
QoS	Quality of Service
RMSEE	Root Mean Square Estimation Error
RN	Reference Node
RSS	Received Signal Strength
RTT	Round Trip Time
Rx	Receiver
SF	Super Frame
SINR	Signal to Interference Plus Noise Ratio
TDMA	Time Division Multiple Access
TDOA	Time Difference of Arrival
TEV	Timed Event
TOA	Time of Arrival
TOF	Time of Flight

UWB	Ultra WideBand
VA	Virtual Anchor
WLAN	Wireless Local Area Network
WLS	Weighted Least Squares
WSN	Wireless Sensor Network

1. Introduction

One of the primary objectives of the WHERE project is to study radio positioning techniques using existing and future heterogeneous communication networks. In order to offer reliable and accurate location information, the WHERE consortium researches novel cooperative positioning schemes as well as the fundamental performance limits.

The concept of cooperative positioning was first proposed in Japan for acquiring real-time positioning information of mobile robots. This concept, mostly applied nowadays to wireless sensor networks (WSNs), has been recently introduced to heterogeneous communication networks. Some of the aspects present in the WSN context are also present in general communication networks, however the heterogeneity of today's wireless communication networks can be seen as an additional problem to be addressed. Current research has aimed at porting WSN positioning algorithms into communication networks. For example in [FRA07], user cooperation was exploited in a least squares framework where cellular and ad-hoc links are combined in a single module of the system. Instead in [FF07], [FIGU08], common Bayesian filtering, namely Kalman filtering, is used for combining short- and long-range links. In [CLTZ07], the authors proposed a mathematical formulation based on the absolute position obtained by the cellular system followed by a routine of optimization that uses the information from the short-range links. Common aspects and subsequently similar results can be expected for example from Cramer-Rao lower bound (CRLB) analysis in WSN. The CRLB was analyzed for some example networks in [LAR04]. In general, some of the important aspects concerning cooperative positioning in wireless networks are for instance: technology integration in a heterogeneous network, positioning information overhead, distributed computing, and node clustering management. Though multihop positioning as it is known in the WSN context is also possible in heterogeneous networks, its implementation may require more robust clustering algorithms since wireless nodes are often mobile. Cooperative positioning techniques can be enhanced at least in the following two aspects:

1) Cooperative positioning techniques proposed for WSN cannot be straightforwardly extended to mobile communication networks. This is because mobile communication networks usually operate in a very complex wireless environment due to many factors such as shadowing, mobility, communication infrastructure, and heterogeneous air-interfaces. Cooperative positioning techniques for various mobile communication scenarios will be addressed mainly in Section 2.

2) Signal-processing complexity and training/signalling overhead are two key problems for existing cooperative positioning schemes. This problem is significant for a wireless network accommodating a large number of mobile nodes. An efficient cooperative positioning scheme should achieve the best trade-off between overhead and performance. This issue will be investigated mainly in Section 2.4, Section 2.5, and Section 3.

Specifically, we investigate the following scenarios that are used to evaluate the performance of cooperative positioning schemes [D11]. The scenario T1.A (as depicted in Figure 1.1) considers the inside of a building in which a meshed network of short-range nodes is deployed. Particularly, this deliverable considers also the multi-hop case from scenario T1.B as depicted in Figure 1.2. Scenario T2.A considers cooperative WiFi complementing cellular communication systems, where this scenario applies mostly for indoor office building. Scenario T3 assumes a typical outdoor urban canyon environment using cellular networks (and a possible hybrid data fusion with GNSS, cf. [D23]).

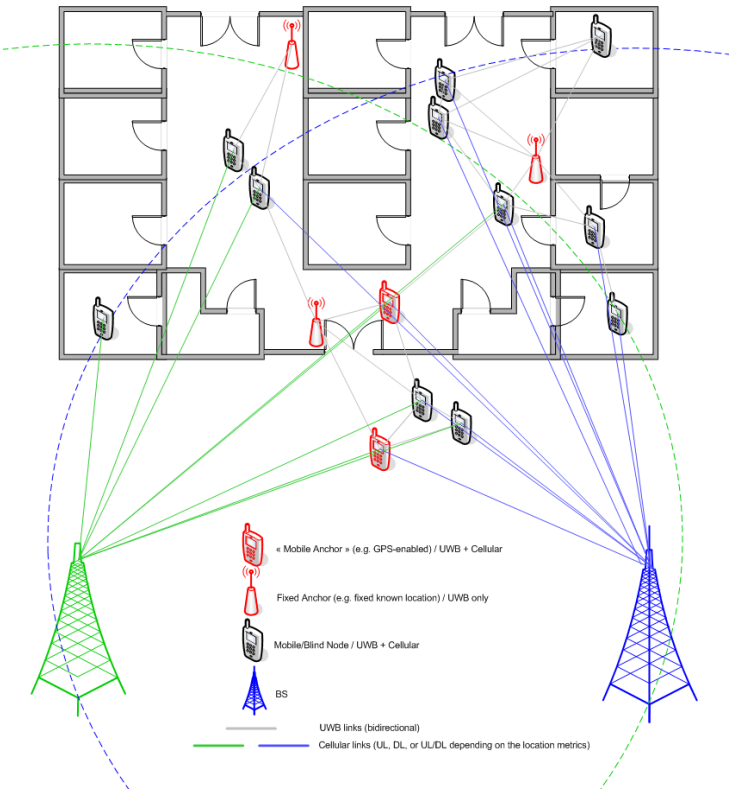


Figure 1.1: Scenario T1.A.

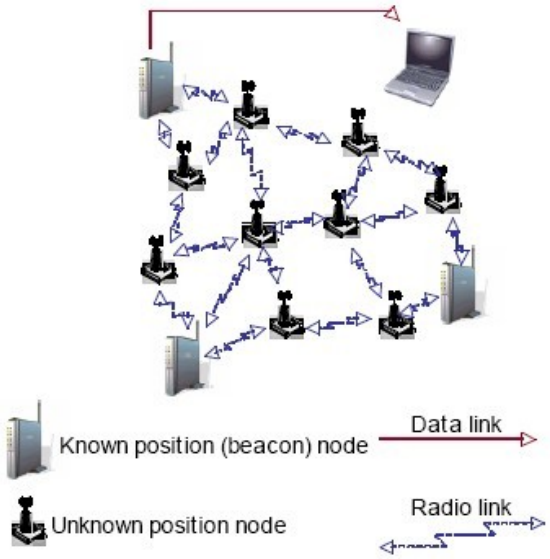


Figure 1.2: Scenario T1-B.

Section 2 presents several novel cooperative positioning schemes for various wireless environments. The focus of Section 2.1, Section 2.2, Section 2.3, Section 2.6, and Section 2.7 is on the scenarios T1.A or T1.B. Specifically, Section 2.1 presents multi-hop cooperative positioning schemes for ad hoc networks. Section 2.2 presents static indoor multi-hop positioning scheme using GBP. In Section 2.3, the mean field algorithm is investigated as an alternative to nonparametric belief propagation algorithms. Section 2.6 presents cooperative positioning using multidimensional scaling. Section 2.7 verifies cooperative positioning experimentally in a typically obstructed indoor scenario, based on real IR-UWB platforms updraded in the frame of WHERE WP5. In addition, Section 2.4 investigates cooperative positioning and tracking algorithms in WiFi (scenario T2.A). Section 2.5 presents cooperative positioning for cellular radio systems (scenario T3).

Section 3 presents the performance analysis for novel cooperative positioning schemes from the communication aspects. Specifically, Section 3.1 investigates effect of delays in WiFi-based (scenario

T2.A) cooperative positioning. Section 3.2 presents new prioritized and decentralized medium access schemes for cooperative tracking with short-range peer-to-peer measurements between mobiles (scenario T1.A).

Finally, we draw the conclusion on the main results and provide an outlook on work ahead.

2. Cooperative Positioning Techniques

2.1 Cooperative Positioning for Ad Hoc Networks

2.1.1 Introduction

Unlike infrastructure networks, ad hoc networks cannot rely on dedicated infrastructure to forward traffic across fixed network segments between mobile users. Furthermore, direct communication between all nodes is infeasible due to limited transmission range. The nodes have to establish multi-hop wireless paths and to cooperate in order to dynamically maintain routes. Cooperation between nodes is essential also for localization, because it allows nodes which are not in range of a sufficient number of references (nodes that are aware of their location), and therefore at the first sight not able to be localized, to be located. In a two-dimensional space, at least three references are required. Given a limited range, it is very unlikely that a node will be able to directly communicate with a sufficient number of references and estimate its position. The proposed framework is transparent to underlying radio access technology and can be applied to WiFi scenarios or 802.15.4-based networks.

2.1.2 Localization in Ad Hoc Networks

The absence of a central unit for network control dictates the need for a distributed solution, where localization is performed by nodes themselves based on pair-wise distance measurements and information collected from local neighbours. This approach significantly reduces energy consumption and computation and communication overhead. A centralized approach may result in uneven power consumption and traffic distribution.

Aspects that have to be considered when choosing or designing an ad hoc positioning algorithm are limited resources, number and density of nodes, percentage of anchors (reference nodes), and network topology. It is also important to choose the appropriate range measurement method for the environment of interest. Limited resources refer mainly to energy constraints in an ad hoc network. Battery lifetime is usually limited, and nodes have low processing capacity. Therefore it is important to have an even distribution of power consumption among nodes, and a distributed approach generally solves this issue. In a purely ad hoc network, not depending on a central server or infrastructure, all of the entities have the same capabilities.

Main performance metrics for cooperative positioning algorithms are position estimation accuracy and latency, whereby the level of accuracy depends on the application for which the system is being used. Latency or response time is an important issue for mobile scenarios. Ability to provide low latency is crucial in dynamic scenarios, so the localization system can be easily updated every time the topology changes. The two major cost parameters are the amount of communication between nodes, and the computation process in the nodes. Cooperative solutions have to achieve desired cost-performance trade-offs. The number of actively participating nodes should be kept at a minimum, and therefore an appropriate cooperation subset has to be chosen, while the other nodes can be ignored. Such a restrictive and selective use of references is crucial in networks with limited resources.

2.1.3 Multi-Hop Localization

The major drawback of least squares based positioning techniques is that they assume that each unknown node is able to communicate directly with anchor nodes and is able to estimate its distances to these anchors. In multi hop schemes nodes have to determine their locations without direct connectivity to anchors, but depending on other localized nodes. These techniques require node collaboration. However, they are more vulnerable because a number of wrongly estimated locations will affect the localization of the remaining ones.

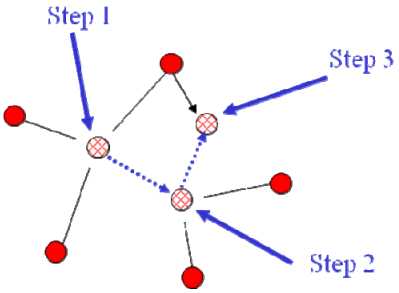


Figure 2.1: Iterative lateration.

The recursive approach, illustrated in Figure 2.1, where the basic algorithm is applied iteratively, by converting newly estimated nodes to anchors, results in error accumulation. As a newly localized node is becoming a new anchor for its neighbours, the estimation error of the first node can propagate to other nodes and eventually get amplified. Over iterations the error could spread throughout the network, leading to abundant error in large topologies. One solution to reduce error accumulation is to use only reliable neighbours as references. Therefore, the algorithm has to assess the quality of distance estimates and integrate that information into the positioning procedure, in order to provide a robust iterative performance. Besides this, it should consider the uncertainties of used anchor nodes. If a newly estimated location has error larger than a predefined threshold, it will be discarded and not used to localize others.

2.1.4 Iterative Framework for Cooperative Positioning

We propose one algorithm based on iterative multilateration, which provides a way to expand the network coverage in a step-by-step fashion. In this sense, coverage is the fraction of nodes that have an accurate position estimate. We assume a middle-scale ad hoc scenario with a certain percentage of anchor nodes and limited radio range. Specifically, we assume the following:

1) $K = 30$ nodes are assumed to be uniformly distributed in a L x L m^2 area ($L = 30$ m).
2) $K_a = 6$ is the number of initial anchor nodes, $K_u = 24$ is the amount of unknown nodes.
3) Transmission range is set to $R = 5$ m, $R = 10$ m and $R = 15$ m, respectively.

Once a node joins the network, it has to perform some sensing to identify which nodes are in its range. During this network discovery phase, signal strength is being measured to decide whether the packet has been received successfully or not. Therefore we decided to use RSS measurements for distance estimation. We use the lognormal shadowing propagation model, with typical parameters for indoor environment. X_{ij} is the measured power P_{ij} at node i transmitted by node j. We assume that P_{ij} is lognormal; therefore, the random variable P_{ij} (dBm) =10log$_{10}$ P_{ij} is Gaussian:

$$\begin{aligned} P_{ij}(\text{dBm}) &\sim N(\overline{P_{ij}}(\text{dBm}), \sigma^2_{\text{dB}}) \\ \overline{P_{ij}}(\text{dBm}) &= P_0(\text{dBm}) - 10\beta \log_{10}(\frac{d_{ij}}{d_0}) \end{aligned} \tag{2.1}$$

where $\overline{P_{ij}}(\text{dBm})$ is the mean power in dBm, σ^2_{dB} is the variance of the shadowing, $P_0(\text{dBm})$ is the received power in dBm at reference distance d_o (usually 1m), β is the path loss exponent, a function of the environment. From [Rap96] the distribution of measured distances d_e for the correct distances d is given as:

$$10\beta \log(\frac{d_e}{d_0}) - 10\beta \log(\frac{d}{d_0}) = X_\sigma [dB] \tag{2.2}$$

where X_σ is a zero-mean Gaussian variable with standard deviation σ, both given in dB. We simulate distance measurements as:

$$d_e = d + d(10^{\frac{X_\sigma}{10\beta}} - 1) \tag{2.3}$$

The second term represents the distance dependent error [SMP03]. This dependence causes larger errors for larger distances, which is consistent with the intuitive assumption from [NN01] that distant nodes are less reliable. Typical parameter values for an indoor scenario are $\beta = 3$ and σ =7. Once the distances to a sufficient number of anchors are obtained, a node has to perform a trilateration/multilateration algorithm to compute its coordinates. In order to enable 2-D positioning, distances to at least three reference points must be available.

If we integrate message exchanges into routing protocols, location discovery is almost free in terms of communication cost. To choose the nodes most suitable for cooperation, we suggest to associate a utility function to the scenario, as a function of all metrics relevant for the positioning algorithm. This function contains useful information, i.e. which nodes are in range of each other, how is the quality of links between them etc. The 'Quality of link' parameter is a representation of the channel condition between two nodes in an ad-hoc environment. It helps to establish a statistical measure of the node-to-anchor channel conditions. To assess the quality of links, we perform 100 RSS measurements, and take the statistical mean and standard deviation. Those sets of measurements, related to specific anchor nodes, with a smaller standard deviation are considered to have better quality, and will have a priority in the reference nodes selection phase.

Once the suitable anchor nodes are chosen and the distances to at least three anchors are estimated, position is calculated using least squares. In the first iteration, a node uses anchors as references, and once it calculates its own position, it becomes a new anchor, but with an associated uncertainty. This uncertainty estimate will also be included in the utility function as a metric for reference selection. According to [APAL06], the quality of links is a more important factor for anchor selection than quality of estimate, and therefore it should have a higher priority as part of the utility function. The framework description is given in the following diagram:

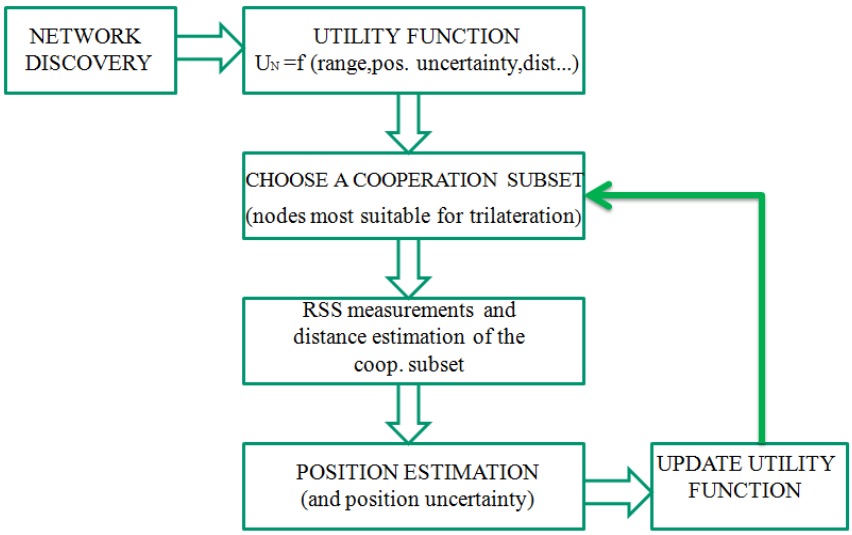


Figure 2.2: Iterative localization framework.

Every newly estimated position acts as a reference point for its neighbourhood and extends system coverage. To decide whether or not a new estimate can advertise itself as an anchor, we need to set an error threshold. Threshold adjustment makes it possible to find a trade-off between desired accuracy and number of iterations needed for full coverage. It is evident that communication range will have an impact on coverage.

To illustrate error propagation, we plot the cumulative distribution function (CDF) of the positioning error after first iteration, where only nodes with exactly known positions are being used as references.
The average localization error $\overline{E_l}$ is defined as:

$$\overline{E_l} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\left(x_i^e - x_i\right)^2 + \left(y_i^e - y_i\right)^2} \qquad (2.4)$$

where N is the total number of localized nodes, (x_i^e, y_i^e) and (x_i, y_i) are the estimated and true position of node *i*, respectively. CDF shows the percentage of localized nodes whose error is smaller than a certain value.

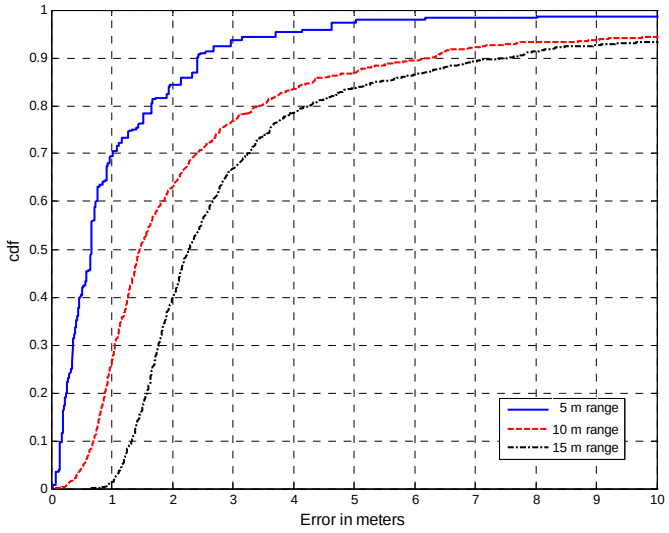


Figure 2.3: Positioning error for first iteration.

The impact of noise is shown in the following figure. We plot the complementary cumulative distribution function for σ = 3dB, 5dB and 7dB, respectively.

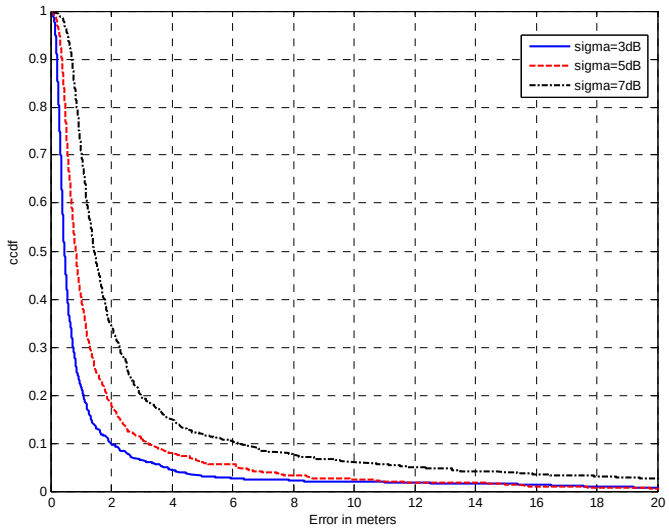


Figure 2.4 Impact of noise on accuracy.

In the second iteration, we use obtained estimates as new anchor nodes. For this reason, the positioning error significantly increases:

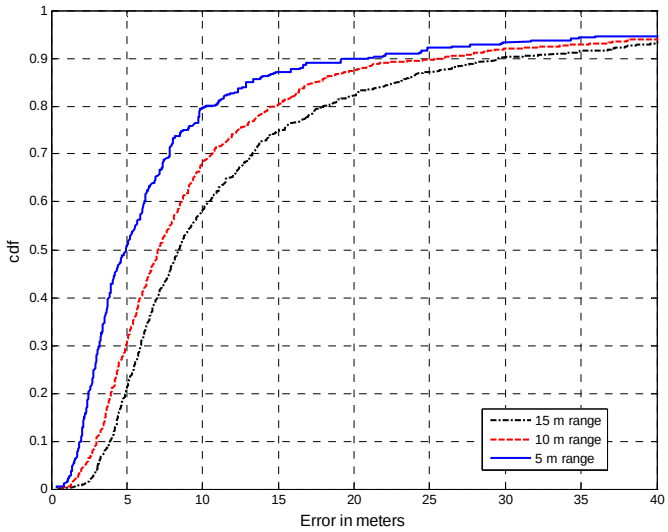


Figure 2.5: Positioning error after second iteration.

To assess the uncertainty of newly estimated anchors, we propose to include a confidence metric as a part of the utility function. During the position estimation phase, the node derives its localization error and converts it into a confidence metric to indicate its position reliability [WYG06].

Let p_e and p be a node's estimated and true position, thus $e = |p_e - p|$ is its error. The function $c = f(e)$ is the label generating function, and c is the confidence label. We assume that c has a value $0 < c < 1$, where $C=1$ is the highest confidence label and corresponds to initial anchor nodes that have exactly known coordinates.

Each node uses the label generating function to convert its localization error to a corresponding label. To construct an appropriate form of the function, we take into account the following:

1) $f(e)$ has to be a decreasing function: if $e_1 > e_2$, then $f(e_1) < f(e_2)$. The more accurate the estimate is, the smaller the error, and confidence increases.
2) The function takes values from a set of values $0 < c < 1$.
3) The domain of the function are the error values, that theoretically can be any value in $(0, +\infty)$.
4) When the error is higher than a threshold value e_0, the output of the function needs to be equal to zero.
5) The form has to be relatively simple and easy to compute.

Having in mind all these properties, we propose the following form:

$$c = f(e) = \max(0, 1 - \frac{e}{e_0}). \tag{2.5}$$

In the same manner we model the inverse label function $f_i(c)$:

$$e_b = f_i(c) = -\frac{e_0}{C} * (c - C) \tag{2.6}$$

The confidence label allows us to filter out unreliable anchors. If e_{max} is the maximum acceptable error for a node to be propagated to an anchor, then $c_0 = f(e_{max})$ is the minimum reasonable confidence label. If c_o is too high, many reference nodes might be filtered out and a node will not be able to obtain distances to a sufficient number of anchors. Therefore it is necessary to find a trade-off between tolerable error and localization coverage. The iterative algorithm with error control is illustrated in Figure 2.6.

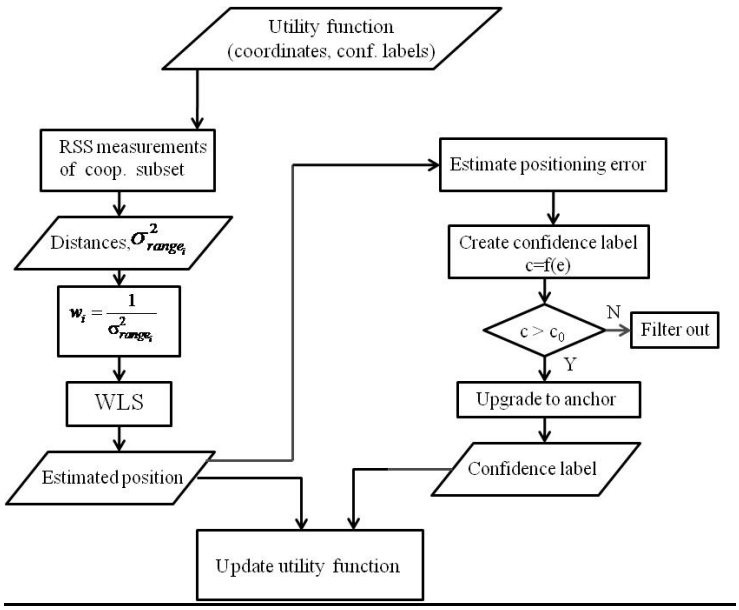


Figure 2.6: Iterative algorithm with error control.

2.1.5 Impact of Anchor Placement

In our algorithm, position calculation is based on the well known least square solution for a set of *n*-1 linear equations:

$$\begin{pmatrix} x_1 - x_n & y_1 - y_n \\ x_2 - x_n & y_2 - y_n \\ \dots & \dots \\ x_{n-1} - x_n & y_{n-1} - y_n \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} d_1^2 - d_n^2 - x_1^2 + x_n^2 - y_1^2 + y_n^2 \\ d_2^2 - d_n^2 - x_2^2 + x_n^2 - y_2^2 + y_n^2 \\ \dots \\ d_{n-1}^2 - d_n^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2 \end{pmatrix} \tag{2.7}$$

which can be rewritten in matrix form as **Ax=b.** Obviously, matrix **A** contains information about anchor geometry, while matrix **b** encodes information about noisy range estimates *d*.

From the expression for the LS solution, it can be seen that there are two factors affecting the estimation error of x: 1) the distance estimation errors resulting from noisy measurements, which are part of the matrix **b**, and 2) the geometry of reference nodes. Information about anchor geometry is part of the matrix **A**. The estimation error is upper bounded [ZHL2006]:

$$\frac{|\Delta \mathbf{x}|}{|\mathbf{x}|} \le cond(\mathbf{A}) \frac{|\Delta \mathbf{b}|}{|\mathbf{b}|} \tag{2.8}$$

For collinear nodes, or if the number of reference nodes is less than three, matrix **A** is singular and its condition number is infinity. For a high value of $cond(\mathbf{A})$ even the minor perturbation in distance estimations would cause a large error in position estimate. If the reference nodes are well separated around the unknown node, matrix **A** is well conditioned. That is why we use $cond(\mathbf{A})$ as a criterion for reference node selection. In our simulations, we rejected all configurations where $cond(\mathbf{A}) > 5$. For higher values of noise it is preferable to choose a smaller threshold, and eventually the threshold could also be set online.

The second term in the equation is due to range measurement error. For a nearby node, the measured RSS value is high compared to the noise, while a distant node has a lower RSS value and thus $\frac{|\Delta b_i|}{|b_i|}$ is large. For this reason it is preferable to use nearby reference nodes.

One advantage of the proposed method is reduced computation. In the error control approach, position is first computed and then the decision is made whether to reject it or not, while here in case of a high condition number, computation does not take place at all. Another advantage is that error control requires

good knowledge of noise characteristics, which is hard to obtain in an unknown environment. However, the error control mechanism prevents errors from propagating throughout the network, while anchor geometry control prevents errors from being generated. The benefit of taking into account anchor geometry for first and second iteration (when using erroneous virtual anchors) is shown on Figure 2.7 and Figure 2.8, respectively.

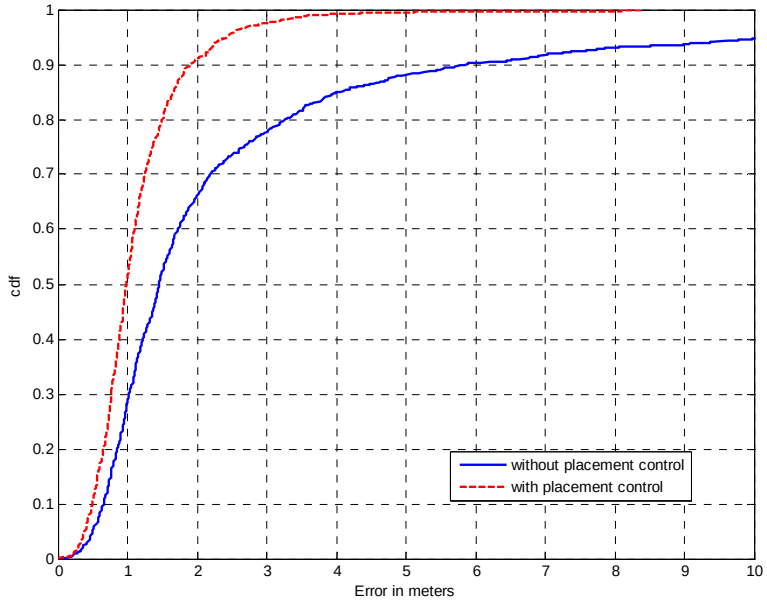


Figure 2.7: Impact of anchor geometry.

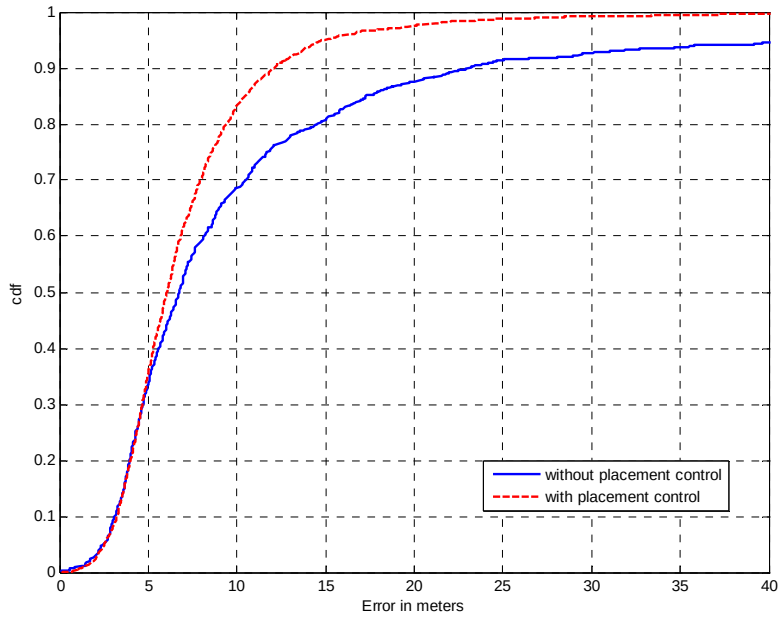


Figure 2.8: Impact of anchor geometry – second iteration.

A selective use of anchors, based on their geometry, improves the positioning accuracy. It is preferable to choose anchors that are well-separated around the node.

2.1.6 Conclusion

In this section, the iterative multilateration approach has been introduced. The method is suitable for ad hoc networks as it is fully distributed, and all computation is performed locally by each node. However, it requires a relatively high degree of connectivity in order to localize all nodes in a small number of iterations. The major drawback of this approach is error propagation and accumulation, resulting from measurement errors, and from the fact that erroneous virtual anchors are used as references. In order to avoid error propagation, a metric for virtual anchor uncertainty has to be modeled. Also anchor selection plays a great role in improving accuracy. Using reference nodes that are well separated helps reduce the positioning error in the least square based calculation.

2.2 Cooperative Positioning using Generalized Belief Propagation Techniques

2.2.1 Introduction

The belief propagation (BP) algorithm is a way of organizing the global computation of marginal beliefs in terms of smaller local computations within the graph. It is one of the best-known graphical models for distributed inference in statistical physics, artificial intelligence, computer vision, error-correcting codes, positioning, etc. The whole computation takes a time proportional to the number of links in the graph, which is significantly less than the exponentially large time that would be required to compute marginal probabilities naively. Due to the presence of nonlinear relationships and highly non-Gaussian uncertainties, especially in indoor sensor positioning, the standard BP is undesirable. In addition, in order to obtain acceptable spatial resolution for the sensors, the discrete space (grid) in the deployment area must be made too large for BP to be computationally feasible [Ihler05]. However, particle-based approximation via *nonparametric belief propagation* (NBP) makes BP acceptable for inference in sensor networks. The main advantages of this approach are easy implementation in a distributed fashion and sufficiency of a small number of iterations to converge. Furthermore, NBP is capable to provide information about location uncertainties and to accommodate non-Gaussian distance measurement errors. Detailed description of NBP and appropriate simulation results are already provided in previous delivery [D22]. Anyway, we will provide the brief description of NBP in following subsection.

However BP&NBP convergence is not guaranteed in a network with loops or even if BP&NBP converges, it could provide us less accurate estimates. Regarding localization using NBP, there is no big convergence issue, but the accuracy is questionable. In the state of the art, there are few solutions for networks with loops [JW02], [YFW03], but mostly they have not been used for the localization. In this work, we review four methods: generalized BP based on Kikuchi approximation (GBP-K), generalized BP based on junction-tree method (GBP-JT), nonparametric GBP-JT (NGBP-JT) and NBP based on spanning trees (NBP-ST).

In standard BP, all messages are always going from a single node to another single node. Obviously, we expect that messages from groups of nodes to other groups of nodes could be more informative, and thus lead to better inference. That is the basic idea behind all techniques of GBP. The mathematical justification of GBP algorithms is that, if we define messages and message-update rules appropriately, we can show that the fixed points of GBP algorithm are equivalent to the stationary points of a corresponding *Kikuchi* approximation to the free energy. In general Kikuchi approximation, the free energy is approximated as a sum of the local free energies of a set of regions of nodes. Therefore, Bethe approximation, used to define standard BP, is just a special case, with two-node "regions" as maximum. GBP based on Kikuchi approximation (GBP-K) algorithm nearly always improves, at least slightly, over the performance of standard BP. However, complexity of GBP-K grows exponentially with the size of the basic clusters that are chosen. But these basic clusters encompass the shortest loops in the graph, which eliminates nearly all the error associated with the BP algorithm. So for many graphical models using such basic clusters (e.g., 3-node clusters) involves only minimally more computation than standard BP. On the other hand, including all loops as the basic clusters, this GBP-K algorithm is exact, but as we already mentioned, computationally unacceptable.

Another method, GBP based on junction trees (GBP-JT), is a standard method for exact inference in graphical models. This can be proved using *elimination* procedure. It is based on *triangulated* graph where "virtual" edges are added so that every loop of length more than 3 has a chord. However, due to the same reason as for NBP, the price is unacceptable large computational cost, so the solution is particle-based version of GBP-JT algorithm (NGBP-JT). NGBP-JT consists of three phases: drawing efficiently particles from the cliques (i.e., high-dimensional particles), computing weighted particles from the messages and then from clique's beliefs.

The previous methods are still very complex for large-scale ad hoc sensor networks. Moreover, the connectivity in these networks is very high which makes computational and communication burdens for low-power applications. Therefore, we are going to break the loops in the network using NBP based on spanning trees (NBP-ST) created by breadth first search (BFS) method. BFS method is optimal method for spanning tree formation in unweighted graphs. Since the NBP method is naturally distributed through the graph (which means that there is no central unit which will handle all computations) the proposed BFS method has to be done in a distributed way. Finally, NBP-ST algorithm represents two (or more) independent runnings of the NBP algorithm based on formed spanning trees. NBP-ST algorithm performs better than NBP in terms of accuracy, computational and communication costs, but only in highly connected (i.e., highly loopy) networks.

Finally, we use real database of measured RSS and TOA obtained in indoor environment, in order to obtain the distance between transmitters. We provide hardware description, RSS/TOA calibration, and the error models. These results will justify the importance of described probabilistic methods since they are capable to handle non-Gaussian uncertainties.

2.2.2 Overview of Localization using Nonparametric Belief Propagation (NBP) Algorithm

We consider the case in which some small number of anchor nodes obtain their coordinates via GPS or by installing them at points with known coordinates, and the rest, unknown nodes, must determine their own coordinates. We suppose that all sensors with unknown positions obtain noisy distance measurements of nearby subset of the other sensors in the network. These measurements can be obtained using a broadcast transmission from each sensor as all other sensors listen. Typical measurements techniques [PAK+05] are time of arrival (TOA), time diference of arrival (TDOA), received signal strength (RSS), and angle of arrival (AOA).

We start by describing a graphical model for the sensor network localization problem. This model is not the most general framework possible, but it's sufficiently flexible to be extended to more complex scenarios. We suppose that all sensors with unknown positions obtain noisy distance measurements of nearby subset of other sensors in the network. This type of problem includes, for example, scenarios in which each sensor is equipped with either a wireless or acoustic transceiver and inter-sensor distances are estimated by measuring the received signal strength or time delay of arrival between sensor locations. Typically, this measurement procedure can be accomplished using a broadcast transmission (acoustic or wireless) from each sensor as all other sensors listen.

Let us assume that we have N_s sensors (N_a anchors and N_u unknowns) scattered randomly in a planar region, and denote the two-dimensional location of sensor t by x_t. The unknown node t obtains a noisy measurement d_{tu} of its distance from node u with some probability $P_d(x_t, x_u)$:

$$d_{tu} = \|x_t - x_u\| + v_{tu}, \qquad v_{tu} \sim p_v(x_t, x_u) \tag{2.9}$$

The binary variable o_{tu} will indicate whether this observation is available or not:

$$o_{tu} = \begin{cases} 1, & d_{tu} \ observed, \\ 0, & otherwise. \end{cases} \tag{2.10}$$

Finally, each sensor t has some prior distribution denoted $p_t(x_t)$. This prior could be an uninformative one for the unknowns and the Dirac impulse for anchors. Then, the joint distribution is given by:

$$p(x_1,...,x_{N_u} | \{o_{tu}\}, \{d_{tu}\}) = \prod_{(t,u)} p(o_{tu} | x_t, x_u) \prod_{(t,u)} p(d_{tu} | x_t, x_u) \prod_t p_t(x_t) \tag{2.11}$$

The final goal of this localization problem is to estimate the maximum a posteriori (MAP) sensor location x_t given a set of observations $\{d_{tu}\}$. There are two different ways to do this, to estimate MAP of each x_t , or to estimate MAP of all x_t jointly. We select the first one because our goal is to minimize the average number of errors instead of minimizing joint probability of error.

The measured distances d_{tu} and d_{ut} may be different, and it is even possible to have $o_{tu} \neq o_{ut}$, indicating that only one of the sensors u and t can observe the other. In this case, we need to symmetrize the observations, i.e., exchange information between any two sensors which observe either d_{tu} or d_{ut}, so that both sensors know the values of all four random variables. This process of exchanging and symmetrizing information may involve multi-hop message routing or other communication protocols. However, we assume that both sensors obtain the same single observation, so $d_{tu} = d_{ut}$ and $o_{tu} = o_{ut}$, otherwise there is no observed distance.

The amount of prior information is also very important. If we want to obtain relative positions, given only the relative measurements $\{o_{tu}, d_{tu}\}$, the sensor location may be solved up to an unknown rotation, translation and negation of the entire network. However, we would like to obtain the absolute coodinates, so we need minimum 3 anchor nodes. The prior distributions of an unknown and anchor node at position x_t^a are respectively given by:

$$p_t^u(x_t) = \begin{cases} 1/S, & \textit{in the deployment area } (S), \\ 0, & \textit{otherwise.} \end{cases}, \quad p_t^a(x_t) = \delta(x_t - x_t^a) \tag{2.12}$$

Of course, unknown nodes may obtain some prior information (high/low probable locations, dependences between unknown nodes,…), so if this information is available, we should include it in previous equation. For large-scale sensor networks, it is reasonable to assume that only a subset of pairwise distances will be available, primarily between sensors which are located within the same radius R. The ideal model of probability of detection is given by:

$$P_d(x_t, x_u) = \begin{cases} 1, & \textit{for } \|x_t - x_u\| \leq R, \\ 0, & \textit{otherwise.} \end{cases} \tag{2.13}$$

However, we use the exponential model, which represents a good approximation of the real-world systems:

$$P_d(x_t, x_u) = \exp\left(-\frac{1}{2} \|x_t - x_u\|^2 / R^2 \right) \tag{2.14}$$

Better approximations of $P_d(x_t, x_u)$ could be done using real experiments in the area of interest, and it is especially advisable for indoor scenarios.

Besides, we have to exchange information between the nodes which are not directly connected. Let's define a pair of nodes s and t to be 1-step neighbors of one another if they observe their pairwise distance d_{st}. Then, we define 2-step neighbors of node s to be all nodes t such that we do not observe the d_{st}, but do observe d_{su} and d_{ut} for some node u. We can follow the same pattern for the 3-step neighbors, and so forth. These n-step neighbors ($n > 1$) contain some information about the distance between them. Therefore, if two nodes do not observe the distance between them, they should prefer to be far away from each other. In our case, we will include all 1-step and 2-step neighbors, while others could be neglected without losing accuracy of the results.

The relationship between the graph and joint distribution may be quantified in terms of potential functions ψ which are defined over each of the graph's cliques:

$$p(x_1, ..., x_{N_u}) \propto \prod_{cliques\ C} \psi_C(\{x_i : i \in C\}) \tag{2.15}$$

We can immediately define potential functions which can express the joint posterior distribution. This only requires potential functions defined over variables associated with single nodes and pairs of nodes. Single-node potential at each node t, and the pairwise potential between nodes t and u, are respectively given by:

$$p(x_1, ..., x_{N_u}, \{o_{tu}\}, \{d_{tu}\}) = \prod_{(t,u)} p(o_{tu} | x_t, x_u) \prod_{(t,u)} p(d_{tu} | x_t, x_u) \prod_t p_t(x_t) \tag{2.16}$$

We expect that some nodes have a higher probability to detect nearby neighbors, so the probability of detection P_d could be given by (2.14). However, uncertainty in the measurement process such as physical

barriers, multipath, and interference results in the fact that sometimes, especially in indoor scenarios, nearby sensors may still not observe each other. Moreover, for the noise distribution p_v, we choose the Gaussian distribution which represents an approximation of the real scenario.

Finally, the joint posterior distribution is given by:

$$p(x_1,...,x_{N_u}|\{o_{tu},d_{tu}\}) \propto \prod_t \psi_t(x_t) \prod_{t,u} \psi_{tu}(x_t,x_u) \tag{2.17}$$

However, we prefer to estimate the posterior marginal distributions for each node t :

$$p(x_t|\{o_{uv},d_{uv}\}) = \int_{x \setminus x_t} p(x_1,...,x_{N_u}|\{o_{tu},d_{tu}\})dx \tag{2.18}$$

Having defined a graphical model, we can now estimate the sensor locations by applying the belief propagation (BP) algorithm. The form of BP as an iterative local message passing algorithm makes this procedure trivial to distribute among the wireless sensor nodes. We apply BP to estimate each sensor's posterior marginal, and use the mean value of this marginal and its associated uncertainty to characterize sensor positions.

Each node t computes its belief $M_t^i(x_t)$, the posterior marginal distribution of two-dimensional position $x_t(x_t^r, x_t^i)$ at iteration i, by taking a product of its local potential ψ_t with the messages from its set of neighbors G_t :

$$M_t^i(x_t) \propto \psi_t(x_t) \prod_{u \in G_t} m_{ut}^i(x_t) \tag{2.19}$$

The messages m_{tu}, from node t to node u, are computed by:

$$m_{tu}^i(x_u) \propto \int \psi_{tu}(x_t,x_u) \frac{M_t^{i-1}(x_t)}{m_{ut}^{i-1}(x_t)} dx_t \tag{2.20}$$

The pseudocode in Table 2.1. describes the initialization and computation phase of this algorithm. To obtain distance measurements, each sensor has to broadcast its ID and to listen for others sensors broadcasts. For any received sensor id, each sensor (except anchors) has to estimate distance between them. In case of $d_{tu} \neq d_{ut}$, the sensor has to communicate with observed neighbors to symmetrize distance measurements (e.g. $d_{tu} = d_{ut} = (d_{tu} + d_{ut})/2$). For tree-like graphs, the number of iterations should be at most the length of the longest path in the graph. However, it's usually su_cient to run until all unknown nodes obtain information from minimum 3 non-colinear anchor nodes.

Initialization:

- For each node obtain local information $p_t(x_t)$, if available
- Obtain distance measurements between all nodes (except the distances between anchors)
- Initialize $m_{ut}^1 = 1$ and $M_t^1 = p_t$ for all u,t

Computation, for each unknown node t :

- Broadcast $M_t^i(x_t)$ to neighbors, and listen for neighbors' broadcasts
- Compute m_{ut}^{i+1} from m_{tu}^i and M_u^i using (2.20)

- Compute new marginal estimate $M_t^{i+1}(x_t)$ using (2.19)
- Repeat previous computation phases until sufficiently converge

Table 2.1: Sensor localization using belief propagation.

The presence of nonlinear relationships and potentially highly non-Gaussian uncertainties in sensor localization makes discrete BP undesirable. Besides, to obtain acceptable spatial resolution for the sensors, the discrete state space must be made too large for BP to be computationally feasible. However, particle-based representations via nonparametric belief propagation (NBP) enables the application of BP to inference in sensor networks [Ihler05].

In NBP, each message is represented using either a sample-based density estimate (e.g. a mixture of Gaussians) or as an analytic function. Both types are necessary for the sensor localization problem. Messages along observed edges (1-step) are represented by samples, while messages along unobserved edges (2-step,…) must be represented as analytic functions since their potentials have the form $1-P_d(x_t,x_u)$ which is typically not normalizable. The belief and message update equations, (2.19) and (2.20), are performed using stochastic approximations, in two stages: first, drawing samples from the belief $M_t^i(x_t)$, then using these samples to approximate each outgoing message m_{tu}^i. We discuss each of these steps in turn, and summarize the procedure with pseudocode in Table 2.2.

Given N weighted samples $\{W_t^{j,i}, X_t^{j,i}\}$ from the belief $M_t^i(x_t)$ obtained at iteration i, we can compute a Gaussian mixture estimate of the outgoing BP message m_{tu}^i. We first consider the case of observed edges between unknown nodes. The distance measurement d_{tu} provides information about how far sensor u is from sensor t, but no information about its relative direction. To draw a sample of the message ($x_{tu}^{j,i+1}$), given the sample X_t^j which represents the position of sensor t, we simply select a direction θ at random, uniformly in the interval $[0,2\pi)$. We then shift X_t^j in the direction of θ by an amount which represents the estimated distance between nodes u and t ($d_{tu}+v^j$):

$$x_{tu}^{j,i+1} = X_t^{j,i} + (d_{tu}+v^j)[\sin(\theta^{j,i})\ \cos(\theta^{j,i})], \quad \theta^{j,i} \sim U[0,2\pi), \quad v^j \sim p_v \tag{2.21}$$

$U[0,2\pi)$ represents the uniform distribution on the interval from 0 to 2π, and v^j is the measurement noise with distribution p_v (e.g., Gaussian). We can now calculate the weight of this sample ($w_{tu}^{j,i+1}$) using (2.20), (2.21), the kernel density estimate (KDE) of potential function, and reasonable approximation of this kernel function with delta impulse:

$$\begin{aligned} m_{tu}^{i+1}(x_u) &\propto \int \psi_{tu}(x_t,x_u)\frac{M_t^i(x_t)}{m_{ut}^i(x_t)}dx_t = \int \psi_{tu}(x_t,x_u)\frac{\sum_j W_t^{j,i}K_t^h(x_t-X_t^{j,i})}{m_{ut}^i(x_t)}dx_t = \\ &\cong \sum_j \frac{W_t^{j,i}}{m_{ut}^i(X_t^{j,i})}\psi_{tu}(X_t^{j,i},x_u) = \sum_j w_{tu}^{j,i+1}k_{tu}^h(x_u-x_{tu}^{j,i+1}) \end{aligned} \tag{2.22}$$

Using last equation and the fact that this message exists only if there is detection between these two nodes (with probability P_d), the weight of sample $x_{tu}^{j,i+1}$ is given by:

$$w_{tu}^{j,i+1} = P_d(X_t^{i,j},x_u)\frac{W_t^{j,i}}{m_{ut}^i(X_t^{i,j})} \tag{2.23}$$

The optimal value for bandwidth h_{tu}^{i+1} could be obtained in a number of possible techniques. The simplest way is to apply the "rule of thumb" estimate:

$$h_{tu}^{i+1} = N^{-\frac{1}{3}}Var(\{x_{tu}^{i+1}\}) \tag{2.24}$$

An important modification to this procedure can be used to improve accuracy and computation cost. Our goal is to accurately estimate belief in the regions of the state space in which it has significant probability mass. Thus, a good proposal distribution is one which allows us to accurately estimate the portions of the

message m_{tu} which overlap these regions of the state space. Our additional information involves utilizing previous iterations' information to determine the angular direction to each of the neighboring sensors. In particular, we use samples from the marginal distribution computed in the previous iteration to form the relative direction θ:

$$\theta^{j,i} = \arctan(X_u^{j,i} - X_t^{j,i}), \qquad i > 1, \quad \theta^{j,i} \in [-\pi, \pi] \tag{2.25}$$

Therefore, in the first iteration ($i = 1$) we calculate θ using (2.21), but starting from second iteration we apply (2.25). This additional information increases the accuracy of this algorithm (as shown in [D22]).

The next task is to obtain messages from anchor nodes to unknown nodes (only observed edges). Of course, it could be done using previous procedure (each sample of the belief would be placed at the known location of the node and weighted by $1/N$), but it will increase computation and communication cost. So we just use the location of the anchor node (x_t^*) to calculate the analytic form of the message. According to equation (2.20), this message is given by:

$$m_{tu}^{i+1}(x_u) \propto \int \psi_{tu}(x_t, x_u) \frac{\delta(x_t - x_t^*)}{m_{ut}^i(x_t)} dx_t = \frac{\psi_{tu}(x_t^*, x_u)}{m_{ut}^i(x_t^*)} \propto \psi_{tu}(x_t^*, x_u) \tag{2.26}$$

Therefore, this message is proportional to potential function which is constant over the iterations and depends only on location of unknown node x_u.

As stated previously, messages along unobserved edges are represented using an analytic function. Using the probability of detection P_d and samples from the belief M_t^i, an estimate of outgoing message to node u is given by:

$$m_{tu}^{i+1}(x_u) = 1 - \sum_j W_t^{j,i} P_d(X_t^{j,i}, x_u) \tag{2.27}$$

Then the messages from the anchor nodes ($W_t^{j,i} = 1/N$) are given by:

$$m_{tu}^{i+1}(x_u) = 1 - P_d(x_t^*, x_u) \tag{2.28}$$

To estimate the belief $M_u^{i+1}(x_u)$ using (2.19), we draw samples from the product of several Gaussian mixture and analytic message. In our case it is very difficult to draw samples from this product, so we use proposal distribution, sum of the Gaussian mixtures, and then re-weight all samples. This procedure is well-known as *mixture importance sampling* [Ihler05].

Denote the set of neighbors of u, having observed edges to u and not including anchors, by G_u^0, and the set of of all neighbors by G_u. In order to draw N samples, we create a collection of kN weighted samples (where $k \geq 1$ is a parameter of the sampling algorithm) by drawing $kN / |G_u^0|$ samples from each message m_{tu} with $t \in G_u^0$ and assigning each sample a weight equal to the ratio:

$$W_u^{j,i+1} = \prod_{v \in G_u} m_{vu}^{i+1} / \sum_{v \in G_u^0} m_{vu}^{i+1} \tag{2.29}$$

Some of these calculated weights are much larger then the rest, especially after more iterations. This means that any sample-based estimate will be unduly dominated by the influence of a few of the particles, and the estimate could be erroneous. To avoid this, we then draw N values independently from collection $\{W_t^{j,i+1}, X_t^{j,i+1}\}$ with probability proportional to their weight, using *resampling with replacement*. This means that we create N equal-weight samples drawn from the product of all incoming messages. A node is located when a convergence criteria is met, e.g., the Kullback-Leibler (KL) divergence can be used, a common measure of difference between two distributions [Ihler05]. The complete procedure, of one iteration of NBP, is shown in Table 2.2

As we already mentioned, the BP/NBP convergence is not guaranteed in a network with loops [Pearl88] or even with convergeence, it could provide us less accurate estimates. Regarding localization using NBP, there is no convergence problem, but the accuracy is questionable [SZ09a, SZ09b]. Therefore, the upcoming sections will be dedicated to this problem.

Compute NBP messages:

Given N weighted samples $\{W_t^{j,i}, X_t^{j,i}\}$ from the belief $M_t^i(x_t)$ construct an approximation to $m_{tu}^{i+1}(x_u)$ for each neighbor $u \in G_t$:

- For $o_{tu} = 1$, approximate the messages between unknown nodes with a Gaussian mixture:
  - Draw random values for θ and v^j :

    $\theta^{j,1} \sim U[0, 2\pi)$, $\theta^{j,i} = \arctan(X_u^{j,i} - X_t^{j,i})$, $v^j \sim p_v$
  - Sample of the message: $x_{tu}^{j,i+1} = X_t^{j,i} + (d_{tu} + v^j)[\sin(\theta^{j,i}) \ \cos(\theta^{j,i})]$
  - Weights: $w_{tu}^{j,i+1} = P_d(X_t^{i,j}, x_u) \dfrac{W_t^{j,i}}{m_{ut}^i(X_t^{i,j})}$
  - Bandwidth: $h_{tu}^{i+1} = N^{-\frac{1}{3}} Var(\{x_{tu}^{i+1}\})$
- For $o_{tu} = 1$, compute messages from the anchors: $m_{tu}^{i+1}(x_u) \propto \psi_{tu}(x_t^*, x_u)$
- For $o_{tu} = 0$, use analytic function:
  - Message from the unknown: $m_{tu}^{i+1}(x_u) = 1 - \sum_j W_t^{j,i} P_d(X_t^{j,i}, x_u)$
  - Message from the anchor: $m_{tu}^{i+1}(x_u) = 1 - P_d(x_t^*, x_u)$

Compute NBP beliefs:

Given several Gaussian mixtures messages $m_{tu}^{i+1} = \left\{x_{tu}^{j,i+1}, w_{tu}^{j,i+1}, h_{tu}^{i+1}\right\}$, $t \in G_u^0$, compute samples from $M_u^{i+1}(x_u)$:

- For each observed neighbor, except anchors, $t \in G_u^0$:
  - Draw $kN / \left|G_u^0\right|$ samples $\{X_u^{j,i+1}\}$ from each messages m_{tu}^{i+1}
  - Weight by $W_u^{j,i+1} = \prod_{v \in G_u} m_{vu}^{i+1} / \sum_{v \in G_u^0} m_{vu}^{i+1}$
- From these kN locations, resample with replacement, to produce N equal-weight samples: $\{X_u^{j,i+1}, 1/N\}$.

Table 2.2: Sensor localization using nonparametric belief propagation (NBP).

2.2.3 Correctness of Belief Propagation

2.2.3.1 Bethe Free Energy

The correctness of BP can be described using *Bethe* approximation to the "free energy" [YFW03, YFW05]. The fixed points of the BP algorithm correspond to the stationary points of the Bethe "free energy". To make this more clear, let's define for one graphical model, a joint probability function $p(\{x\})$. If we have some other approximate joint probability function $b(\{x\})$, we can define a "distance" between $p(\{x\})$ and $b(\{x\})$, called Kullback-Leibler (KL) distance, by:

$$D(b(\{x\})\,\|\,p(\{x\})) = \sum_{\{x\}} b(\{x\}) \ln \frac{b(\{x\})}{p(\{x\})} \tag{2.30}$$

The KL distance is useful because it is always non-negative and is zero if and only if the two probability functions $p(\{x\})$ and $b(\{x\})$ are equal.

Statistical physicists generally assume that Boltzmann's law is true:

$$p(\{x\}) = \frac{1}{Z} e^{-E(\{x\})/T} \tag{2.31}$$

where Z is a normalization constant, and the "temperature" T is just a parameter that defines a scale of units for the "energy" E. For simplicity, we can choose $T = 1$. Using (2.30) and (2.31), we find the KL distance:

$$D(b(\{x\})\,\|\,p(\{x\})) = \sum_{\{x\}} b(\{x\}) E(\{x\}) + \sum_{\{x\}} b(\{x\}) \ln b(\{x\}) + \ln Z \tag{2.32}$$

So we see that this KL distance will be zero when approximate probability function $b(\{x\})$ will be equal to the exact probability function $p(\{x\})$. The Bethe approximation is the case when joint belief $b(\{x\})$ is function of single-node beliefs $b(x_i)$ and two-node beliefs $b(x_i, x_j)$. Yedidia *et al.* proved in [YFW03] that for a single-connected graph, values of these beliefs that minimize the Bethe free energy, will correspond to the exact marginal probabilities. For graph with loops, these beliefs will only be approximations, although a lot of them are quite good.

2.2.3.2 Double Counting Problem

The problem in networks with loops can be also intuitively explained. Let us consider the example of network in Figure 2.9. In this network, there are 3 unknown nodes (*A*, *B* and *C*) and 3 anchor nodes (E_A, E_B, and E_C) which represent the local evidence. The message-passing algorithm (BP) can be thought of as a way of communicating local evidences between nodes such that all nodes calculate their beliefs given all the evidence.

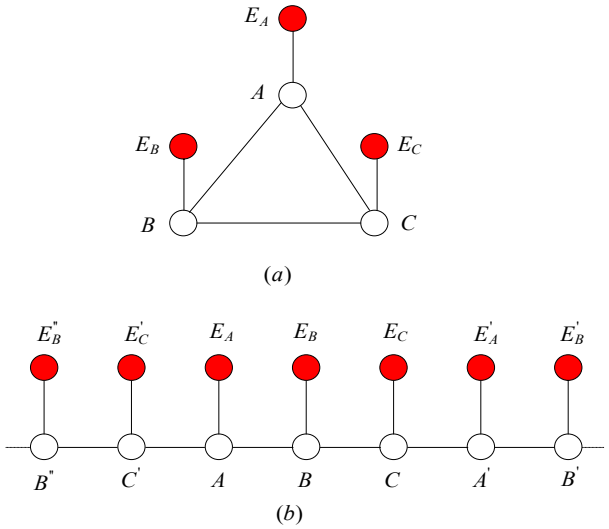


Figure 2.9 (a) A simple loopy network, (b) Corresponding unwrapped network for the first 3 iterations

In order for BP to be successful, it needs to avoid *double counting* [Pearl88, Weiss00] - a situation in which the same evidence is passed around the network multiple times and mistaken for new evidence. Of course, this is not possible in single-connected network because when node receives some evidence, it will never receive that evidence again. In a loopy network double counting could not be avoided. For example in Figure 2.9a, node B will send A's evidence to C, but in the next iteration, C will send that same information back to A. Thus it seems that BP in such network will always give the wrong answer.

However, BP could still lead to correct inference if all evidence is "double counted" in equal amounts. This could be formalized by unwrapped network corresponding to as loopy network. The unwrapped network is a single-connected network constructed such that performing BP in the unwrapped network is equivalent to performing BP in the loopy network. The basic idea is to replicate the nodes as shown in Figure 2.9b. For example, the message received by node B after 3 iterations of BP in the loopy network are identical to the final messages received by node B" in the unwrapped network. In this way, we can create infinite network. The importance of the unwrapped network is that since it is single-connected, BP on it is guaranteed to give the correct beliefs. However, usefulness of this beliefs depends on the similarity between the probability distribution induced by the unwrapped problem and the original loopy problem. And this similarity is satisfied in single-loop network after finite number of iterations. In general case, BP will converge when addition of these additional nodes at the boundary will not alter the posterior probability of the node in the center.

2.2.3.3 Gaussian Networks

Finally, according to [WF01], we review the correctness of BP in Gaussian networks:

- In all single loop Gaussian networks fast convergence is correlated with good approximation of the beliefs.
- In Gaussians and in single loop discrete networks the factor that determines the goodness of approximation and the convergence rate is the amount of statistical independence between the root nodes and the leaf nodes in the unwrapped network.
- In Gaussian networks with multiple loops the mean at each node is guaranteed to be correct but the confidence around that mean may be incorrect.

These results give a theoretical justification for applying BP in certain networks with multiple loops. This may enable fast, approximate probabilistic inference in a range of new applications, including localization where measurement error is usually (but not always) similar to the Gaussian model.

For extensive analysis of this topic, we refer the reader to [Ihler05, MK07] where a lot of useful theorems are provided.

2.2.4 Generalized Belief Propagation Based on Kikuchi Approximation (GBP-K)

2.2.4.1 Definition

In standard BP, all messages are always going from a single node to another single node. It is natural to expect that messages from groups of nodes to other groups of nodes could be more informative, and thus lead to better inference. That is the basic idea behind all techniques of GBP.

The mathematical justification of GBP algorithms is that, if we define messages and message-update rules appropriately, we can show that the fixed points of GBP algorithm are equivalent to the stationary points of a corresponding *Kikuchi* approximation to the free energy. In general Kikuchi approximation, the free energy is approximated as a sum of the local free energies of a set of regions of nodes. Therefore, Bethe approximation (see Section 2.2.3.1), used to define standard BP, is just special case with 2-node "regions" as maximum. In Figure 2.10, we show the basic clusters for both approximations.

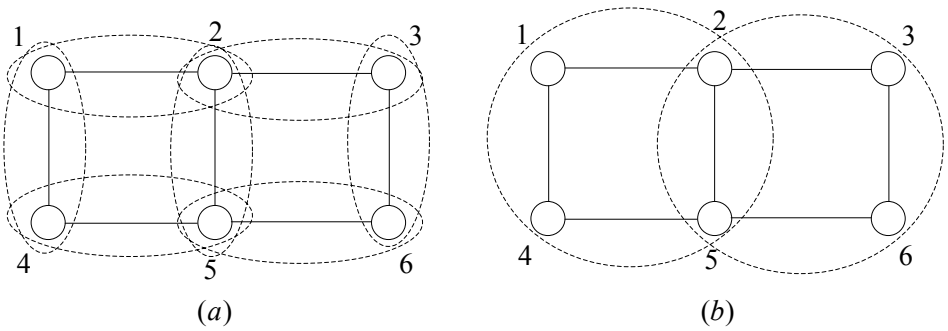


Figure 2.10 The basic clusters in the (a) Bethe approximation and (b) Kikuchi approximation.

In fact, just as BP can be defined without reference to the Bethe approximation, GBP could be defined without referring directly to the Kikuchi approximation. For detailed explanation of Kikuchi approximation, the reader is referred to [YFW03]. Thus, we will now show how this version of GBP based on Kikuchi approximation (GBP-K), also called *region graph method*, can be implemented using a simple example.

2.2.4.2 Example Network

Let us start with Figure 2.11 and try to construct a GBP-K algorithm that correspond to the mentioned Kikuchi approximation. The basic clusters have four nodes each: {1,2,4,5}, {2,3,5,6}, {4,5,7,8}, and {5,6,8,9}. The first step in constructing the algorithm is to find all the intersection regions of the basic clusters, and all their intersection regions, and so on. We find intersection regions {2,5}, {4,5}, {5,6}, and {5,8}, and the single region that is intersection of intersections {5}.

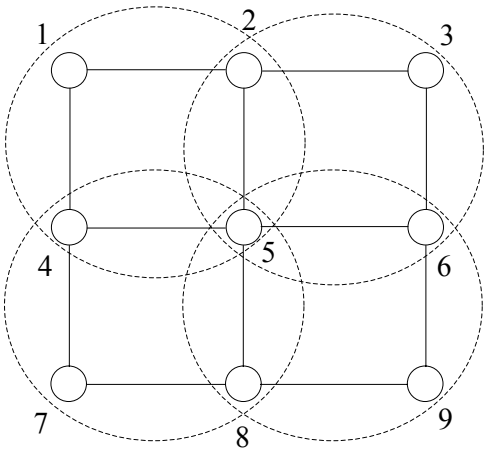
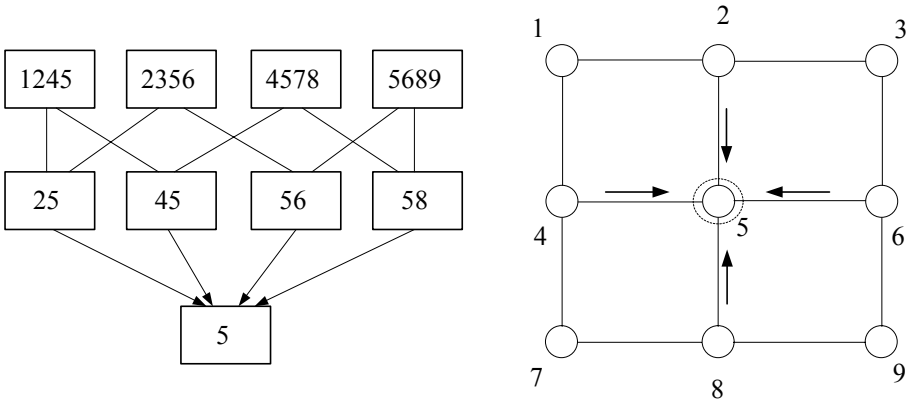


Figure 2.11 Four-node clusters in a 9-node network.

The next step is to organize all the regions into a region graph: a hierarchy of regions and their direct sub-regions (Figure 2.12a). For example, region {5} is not direct sub-region of region {1,2,4,5}, because it is also a sub-region of region {2,5}.



(*a*) (*b*)

Figure 2.12 (a) Region graph for the network from Figure 2.11 (b) Computing belief for the region consisting just of node 5.

Now we can start to construct messages connecting all regions to all their direct subregions and beliefs of the regions. Consider for example the message connecting region {1,2,4,5} to region {2,5}. This is the message from nodes 1 and 4 to nodes 2 and 5, so we denote this message by $m_{14\to 25}(x_2, x_5)$ and for notation simplicity we write it as $m_{14\to 25}$. In general, we can consider a message connecting a region *r* and a sub-region *s* to be a message from those nodes in *r*, that are not in *s*, to the nodes in *s*.

We construct belief equations for every region according to the rule that the belief is proportional to the product of every evidence contained in the region, and every message that goes into nodes in the region from nodes outside the region. For example, for the region {5} (Figure 2.12b), consisting just of node 5, the belief equation is:

$$b_5 = k\psi_5 m_{2\to 5} m_{4\to 5} m_{6\to 5} m_{8\to 5} \tag{2.33}$$

where *k* is a normalization constant and ψ_5 is the potential function of region {5}. Taking the region {4,5} as an example of a 2-node region, and the region {1,2,4,5} as an example of 4-node region, the belief equations are respectively given by:

$$b_{45} = k\psi_4\psi_5\psi_{45} m_{12\to 45} m_{78\to 45} m_{2\to 5} m_{6\to 5} m_{8\to 5} \tag{2.34}$$

$$b_{1245} = k\psi_1\psi_2\psi_4\psi_5\psi_{12}\psi_{14}\psi_{25}\psi_{45} m_{36\to 25} m_{78\to 45} m_{6\to 5} m_{8\to 5} \tag{2.35}$$

The next step in constructing a GBP-K algorithm is to enforce marginalization condition relating each pair of regions that are connected in the hierarchy shown in Figure 2.12a. For example, the marginalization condition which connects the region {5} with the region {4,5} is $b_5(x_5) = \sum_{x_4} b_{45}(x_4, x_5)$. If we combine that with equations (2.33) and (2.34), we find, by cancelling the common terms, the message update rule:

$$m_{4\to 5} = k \sum_{x_4} \psi_4\psi_{45} m_{12\to 45} m_{78\to 45} \tag{2.36}$$

Other message update rules can be found using the analog procedure. The collection of all belief equations and message update rules defines GBP-K algorithm. A GBP-K runs in the same way as the BP algorithm. One normally initializes all the messages to their unbiased states, and then iterates the message update rules until they (hopefully) converge. When convergence of the messages is achieved, the desired beliefs can be read off from the belief equations.

GBP-K algorithm nearly always improves, at least slightly, over the performance of standard BP, and it can significantly outperform standard BP if the graphical model under consideration has short loops. However, complexity of GBP-K grows exponentially with the size of the basic clusters that are chosen. But these basic clusters encompass the shortest loops in the graph, which eliminates nearly all the error associated with the BP algorithm. Thus, for many graphical models using such basic clusters (e.g. 3-node clusters) involves only minimally more computation than standard BP. On the other hand, including all loops as the basic clusters, this GBP-K algorithm is exact, but as we already mentioned, computationally unacceptable.

2.2.5 Generalized Belief Propagation Based on Junction-tree Method (GBP-JT)

2.2.5.1 Definition

In this section, we describe new approach for localization in loopy networks, generalized belief propagation based on junction tree method (GBP-JT) [SZ09a]. This algorithm could be used for localization as well as for significant number of other applications.

Junction tree algorithm is a standard method for exact inference in graphical model. This can be proved using *elimination* procedure [JW02]. The graph is first *triangulated* (i.e. added "virtual" edges so that every loop of length more than 3 has a chord). Given a triangulated graph, with cliques C_i and potentials $\psi_{C_i}(x_{C_i})$, and given the corresponding junction tree which defines links between the cliques, we send the following message from clique C_i to clique C_j by the message update rule:

$$m_{ij}(x_{S_{ij}}) = \sum_{C_i \setminus S_{ij}} \psi_{C_i}(x_{C_i}) \prod_{k \in G_i \setminus j} m_{ki}(x_{S_{ki}}) \tag{2.37}$$

where $S_{ij} = C_i \cap C_j$, and where G_i are the neighbors of clique C_i in the junction tree. The belief at clique C_i is proportional to the product of the local evidence at that clique and all the messages coming into clique *i*:

$$M_i(x_{C_i}) = k\psi_{C_i}(x_{C_i}) \prod_{j \in G_i} m_{ji}(x_{S_{ji}}) \tag{2.38}$$

Beliefs for single nodes can be obtained via further marginalization:

$$M_i(x_i) = \sum_{C_i \setminus i} M_i(x_{C_i}) \text{ for } i \in C_i \tag{2.39}$$

Equations (2.37), (2.38), and(2.39) represent GBP-JT algorithm which is valid for arbitrary graphs. The standard BP algorithm is a special case of GBP-JT, obtained by noting that the original tree is already triangulated, and has only pairs of nodes as cliques. In that case, sets S_{ij} are single nodes, and marginalization using (2.39) is unnecessary.

2.2.5.2 Example Network

Let us show how it works in our example in
Figure 2.13. The network has 10 nodes, 5 anchors (nodes 6-10) and 5 unknowns (nodes 1-5). There is a loop 1-2-4-5-3, so we have to triangulate it by adding two more edges (2-3 and 3-4). Then we define 8 cliques in the graph: $C_1 = \{x_1, x_2, x_3\}$, $C_2 = \{x_2, x_3, x_4\}$, $C_3 = \{x_3, x_4, x_5\}$, $C_4 = \{x_4, x_9\}$, $C_5 = \{x_5, x_{10}\}$, $C_6 = \{x_1, x_6\}$, $C_7 = \{x_2, x_7\}$, $C_8 = \{x_3, x_8\}$. The appropriate potentials of 3-node cliques are given by:

$$\psi_{C_1}(x_1, x_2, x_3) = \psi_{12}(x_1, x_2)\psi_{13}(x_1, x_3) \tag{2.40}$$

$$\psi_{C_2}(x_2, x_3, x_4) = \psi_{24}(x_2, x_4) \tag{2.41}$$

$$\psi_{C_3}(x_3, x_4, x_5) = \psi_{35}(x_3, x_5)\psi_{45}(x_4, x_5) \tag{2.42}$$

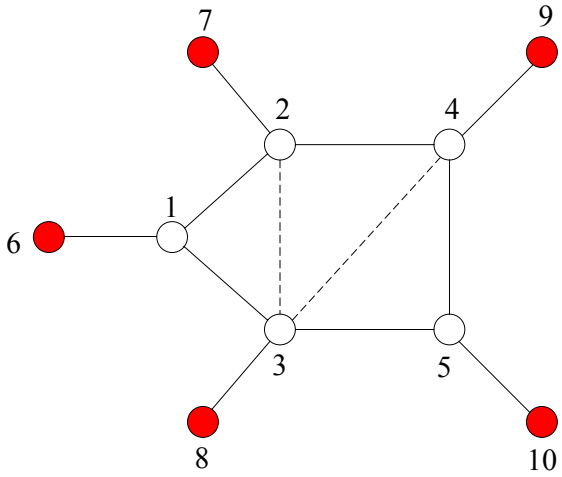


Figure 2.13 Example of 10-node network with loop with 5 anchors (nodes 6-10), and 5 unknowns (nodes 1-5). The network is already triangulated by adding 2 more edges (marked by dashed lines).

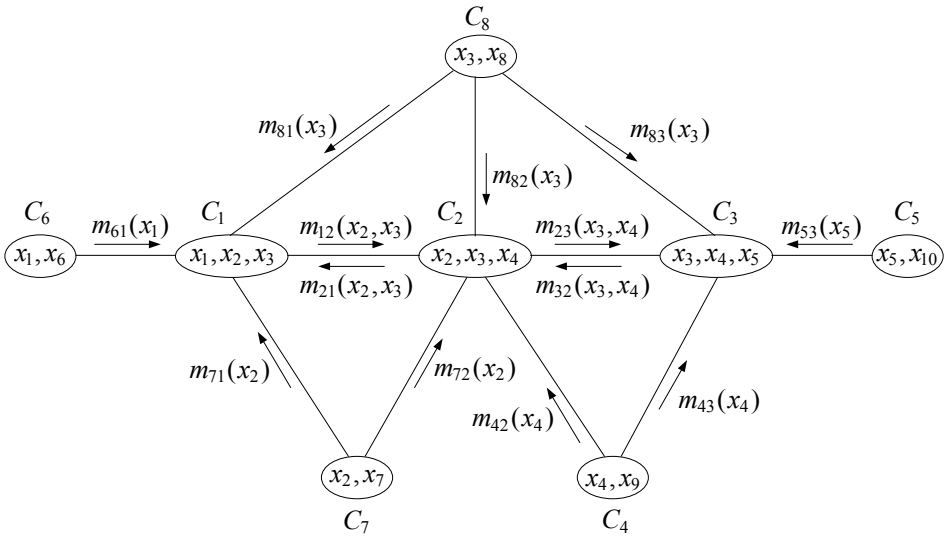


Figure 2.14 The junction tree corresponding to the network in Figure 2.13.

Note that "virtual" edges do not appear in these equations since they are used only to define cliques. Other cliques, defined over pairs of nodes, are nothing else than potential functions between two nodes already known from standard BP:

$$\psi_{C_4}(x_4, x_9) = \psi_{49}(x_4, x_9), \psi_{C_5}(x_5, x_{10}) = \psi_{510}(x_5, x_{10}),$$
$$\psi_{C_6}(x_1, x_6) = \psi_{16}(x_1, x_6), \psi_{C_7}(x_2, x_7) = \psi_{27}(x_2, x_7), \psi_{C_8}(x_3, x_8) = \psi_{38}(x_3, x_8). \quad (2.43)$$

The junction tree corresponding to the network in Figure 2.13 is shown in Figure 2.14. As we can see, "anchor cliques" ($C_4 - C_8$) do not receive messages, so this graph does not contain loops. Actually, these "anchor cliques" also include one unknown node so we can send them messages, but this node also could be located by marginalizing the belief of some other clique. In the next step, we can compute all messages using (2.37). The complete set of messages is given by:

$$m_{61}(x_1) = \psi_{16}(x_1, x_6^*), m_{53}(x_5) = \psi_{510}(x_5, x_{10}^*), m_{71}(x_2) = m_{72}(x_2) = \psi_{27}(x_2, x_7^*),$$
$$m_{42}(x_4) = m_{43}(x_4) = \psi_{49}(x_4, x_9^*), m_{81}(x_3) = m_{82}(x_3) = m_{83}(x_3) = \psi_{38}(x_3, x_8^*) \quad (2.44)$$

$$m_{12}(x_2,x_3)=\psi_{27}(x_2,x_7^*)\psi_{38}(x_3,x_8^*)\sum_{x_1}\psi_{16}(x_1,x_6^*)\psi_{C_1} \quad (2.45)$$

$$m_{32}(x_3,x_4)=\psi_{49}(x_4,x_9^*)\psi_{38}(x_3,x_8^*)\sum_{x_5}\psi_{510}(x_5,x_{10}^*)\psi_{C_3} \quad (2.46)$$

$$m_{21}(x_2,x_3)=\psi_{27}(x_2,x_7^*)\psi_{38}(x_3,x_8^*)\sum_{x_4}\psi_{49}(x_4,x_9^*)\psi_{C_2}m_{32} \quad (2.47)$$

$$m_{23}(x_3,x_4)=\psi_{49}(x_4,x_9^*)\psi_{38}(x_3,x_8^*)\sum_{x_2}\psi_{27}(x_2,x_7^*)\psi_{C_2}m_{12} \quad (2.48)$$

where asterisk denotes the known location of the anchor node and the messages from "anchor cliques" are directly replaced by the appropriate potential function. Moreover, we used simplified notation for messages and clique potentials on the right side of equations (e.g. $m_{12}=m_{12}(x_2,x_3)$, $\psi_{C_2}=\psi_{C_2}(x_2,x_3,x_4)$).

The beliefs of cliques are computed using (2.38):

$$M_1(x_1,x_2,x_3)=\psi_{C_1}\psi_{16}(x_1,x_6^*)\psi_{27}(x_2,x_7^*)\psi_{38}(x_3,x_8^*)m_{21} \quad (2.49)$$

$$M_2(x_2,x_3,x_4)=\psi_{C_2}\psi_{27}(x_2,x_7^*)\psi_{38}(x_3,x_8^*)\psi_{49}(x_4,x_9^*)m_{12}m_{32} \quad (2.50)$$

$$M_3(x_3,x_4,x_5)=\psi_{C_3}\psi_{38}(x_3,x_8^*)\psi_{49}(x_4,x_9^*)\psi_{510}(x_5,x_{10}^*)m_{23} \quad (2.51)$$

Now it's easy to compute beliefs of single nodes by marginalizing beliefs of cliques using (2.39). Obviously, it's sufficient to know beliefs of C_1 and C_3 since these cliques include all unknown nodes. Marginalization of C_2 provides a degree of freedom and could be used to check the estimated positions of some nodes (in our case, for nodes 2, 3 and 4).

The proposed GBP-JT algorithm is not unique. There are a lot of variations of this method; the best known is *cluster variation method* [YFW05]. The main goal is achieved in all of them: estimated beliefs are correct in networks with loops. However, due to the same reason as for NBP, the price is unacceptable large computational cost, so we are going to implement approximated version of GBP-JT algorithm.

2.2.6 Nonparametric Generalized Belief Propagation Based on Junction-tree Method (NGBP-JT)

Due to the same reasons described in previous work [D22] as well as in previous section (computational problems, presence of nonlinear relationships and highly non-Gaussian uncertainties), in this section we propose NGBP-JT [SZ09b], particle-based approximation of GBP-JT method, for the same example of network from the previous section (
Figure 2.13 and Figure 2.14). In order to use this method for localization, we use the same model as for NBP (see Section 3.2.2) (measured distance, probability of detection and potential functions). In addition, we assume that we didn't obtain a priori information about node position, so single-node potentials are equal to 1 (otherwise, beliefs computed using (2.38) must be multiplied by their own potentials).

2.2.6.1 Drawing Initial Particles

Let us draw N_C weighted particles from cliques C_1 and C_3:

$$\{W_1^j,X_1^j\}=\{W_1^j,[X_{1,1}^j,X_{1,2}^j,X_{1,3}^j]\} \quad (2.52)$$

$$\{W_3^j,X_3^j\}=\{W_3^j,[X_{3,3}^j,X_{3,4}^j,X_{3,5}^j]\} \quad (2.53)$$

where W_m^j represents the weight of 6-dimensional (6D) particle X_m^j from clique C_m which consists from three 2-dimensional (2D) particles from node t ($X_{m,t}^j$). For now, we don't need any particle from clique C_2 since they will be computed (not drawn!) in message-update phase. There is a lot of ways to draw these particles. In general, we can draw all particles uniformly within the deployment area, but it requires significant number of particles (e.g. 100 particles drawn from each node, corresponds to $100\times100\times100=10^6$ particles from its clique). Therefore, we immediately include all information

available within the clique: potential functions and (2.40) - (2.42) which represent our information about distance between nodes within the clique. In fact, these potentials represent our importance density function from we draw the particles. First, we draw particles from node *t* uniformly within the deployment area. To draw a particle from any neighboring node *u*, we shift the particle from node *t* in a random direction for an amount which represents the observed distance between these two nodes:

$$X_{m,u}^{j} = X_{m,t}^{j} + (d_{tu} + v)[\sin(\theta^{j})\ \cos(\theta^{j})] \tag{2.54}$$

where $\theta^{j} \sim \text{Unif}[0,2\pi)$, $j = 1,...,N_C$. We will use simplified notation of the above equation:

$$X_{m,u}^{j} = \text{shift}(X_{m,t}^{j}, d_{tu}) \tag{2.55}$$

Assuming that we have already drawn particles, e.g. from nodes 1 and 5, we can compute particles from other nodes:

$$X_{1,2}^{j} = \text{shift}(X_{1,1}^{j}, d_{12}), X_{1,3}^{j} = \text{shift}(X_{1,1}^{j}, d_{13}) \tag{2.56}$$

$$X_{3,4}^{j} = \text{shift}(X_{3,5}^{j}, d_{45}), X_{3,3}^{j} = \text{shift}(X_{3,5}^{j}, d_{35}) \tag{2.57}$$

Since these particles are drawn from ψ_{C_1} and ψ_{C_3} respectively, and we already included all information which place these particles in high-probabilistic regions with respect to $X_{1,1}^{j}$ and $X_{3,5}^{j}$ (see (2.56), (2.57)), all clique's weights can be approximated with the same value:

$$W_1^{j} = W_3^{j} = \frac{1}{N_C}, j = 1,...,N_C \tag{2.58}$$

Note that all particles from nodes within the clique have one common weight, e.g. $\{W_1^{j},[X_{1,1}^{j}, X_{1,2}^{j}, X_{1,3}^{j}]\}$ since we are implementing algorithm which operates with *hypernodes* (cliques). Our initial set of particles from clique C_1 is illustrated in Figure 2.15.

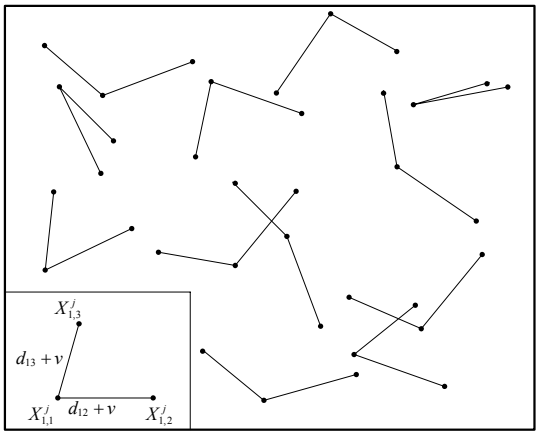


Figure 2.15: Initial set of particles from clique C_1 . The particles from the clique are randomly distributed across the deployment area.

2.2.6.2 Computing Messages

Having drawn all particles, we can now compute all messages. Messages m_{23} and m_{21} are function of m_{12} and m_{32} respectively (see (2.48) and (2.47)), so they will be computed after computing these two messages. Also, messages from the "anchor cliques" will be directly replaced with appropriate potential functions. So we start with messages m_{12} and m_{32} which depends on ψ_{C_1} and ψ_{C_3} from which we have already drawn particles. Let us represent these two messages in slightly different form:

$$m_{12}(x_2,x_3)=\sum_{x_1}M_{12}(x_1,x_2,x_3)$$

$$M_{12}(x_1,x_2,x_3)=\psi_{27}(x_2,x_7^*)\psi_{38}(x_3,x_8^*)\psi_{16}(x_1,x_6^*)\psi_{C_1} \quad (2.59)$$

$$m_{32}(x_3,x_4)=\sum_{x_5}M_{32}(x_3,x_4,x_5)$$

$$M_{32}(x_3,x_4,x_5)=\psi_{49}(x_4,x_9^*)\psi_{38}(x_3,x_8^*)\psi_{510}(x_5,x_{10}^*)\psi_{C_3} \quad (2.60)$$

Defined factors, M_{12} and M_{22}, are some kind of unmarginalized messages, so we will call them *joint messages*. Now it's very easy to compute weighted particles from these joint messages ($\{W_{mn}^j, X_{mn}^j\}$):

$$X_{12}^j = X_1^j = [X_{1,1}^j, X_{1,2}^j, X_{1,3}^j], W_{12}^j = \psi_{27}(X_{1,2}^j, x_7^*)\psi_{38}(X_{1,3}^j, x_8^*)\psi_{16}(X_{1,1}^j, x_6^*)W_1^j \quad (2.61)$$

$$X_{32}^j = X_3^j = [X_{3,3}^j, X_{3,4}^j, X_{3,5}^j], W_{32}^j = \psi_{49}(X_{3,4}^j, x_9^*)\psi_{38}(X_{3,3}^j, x_8^*)\psi_{510}(X_{3,5}^j, x_{10}^*)W_3^j \quad (2.62)$$

Before computing final messages, we noticed the same problem as for NBP, *sample depletion* [DKZ+03, Ihler05], the problem when one, or few, of the weights are much larger than the rest. In our case, it is expected because we are working in 6D space where it's very hard to draw good sample (clique with position and shape similar to the right one - see Figure 2.15). Therefore, we resample with replacement, which will produce N_C equal-weight particles ($W_{mn}^j = 1/N_C$). In our case, we have to resample from cliques, thus the easiest way is to resample from single nodes using standard resampling procedure [AMG+02], and then to synchronize subscripts in order to keep original shapes of the particles. This procedure is illustrated for M_{12}, by the following pseudocode:

Algorithm 2.1: Resample and synchronize.

1: $[W_{12},\ X_{1,1}, index] = \text{resample}(W_{12}, X_{1,1})$
2: **for** $j = 1 : N_C$ **do**
3: $\quad X_{1,2}^j = X_{1,2}^{index(j)};\ X_{1,2}^j = X_{1,2}^{index(j)}$
4: **end for**

where $\{W_{12}, X_{1,1}\}$ is the vector of N_C particles from node 1 (part of the joint message), and *index* is the vector of old (pre-resampled) subscripts of new particles.

Now we are ready to compute particles from messages ($\{w_{mn}^j, x_{mn}^j\}$). The marginalization of joint messages is straightforward since we already have weighted particles from them. So we just need to discard one data, and keep the same weights. Thus, they are given by:

$$x_{12}^j = X_{12}^j(2:3) = [X_{1,2}^j, X_{1,3}^j], w_{12}^j = W_{12}^j = 1/N_C \quad (2.63)$$

$$x_{32}^j = X_{32}^j(1:2) = [X_{3,3}^j, X_{3,4}^j], w_{32}^j = W_{32}^j = 1/N_C \quad (2.64)$$

Finally, we can compute particles of other two messages, m_{23} and m_{21}. According to (2.48) and (2.47), they are function of $\psi_{C_2} m_{12}$ and $\psi_{C_2} m_{32}$, respectively, so we will draw particles from these products and then re-weight by the remainder of (2.48) and (2.47). Actually, two single-node particles of messages m_{12} and m_{32} are already computed (see (2.63) and (2.64)), so we have just to draw the missing particle using information from ψ_{C_2}, the observed distance between nodes 2 and 4. The result of this procedure are particles of joint messages M_{21} and M_{23}. By marginalizing them, we obtain final messages m_{21} and m_{23}. The complete procedures are given as follows:

$$X_{21}^j = [\text{shift}(x_{32}^j(2), d_{24}), x_{32}^j(1), x_{32}^j(2)]$$
$$W_{21}^j = \psi_{49}(X_{21}^j(3), x_9^*)\psi_{38}(X_{21}^j(2), x_8^*)\psi_{27}(X_{21}^j(1), x_7^*)w_{32}^j$$
$$-\textit{resample and synchronize}$$
$$x_{21}^j = X_{21}^j(1:2) = [X_{21}^j(1), X_{21}^j(2)], w_{21}^j = W_{21}^j = 1/N_C \quad (2.65)$$

$$X_{23}^j = [x_{12}^j(1), x_{12}^j(2), \text{shift}(x_{12}^j(1), d_{24})]$$
$$W_{23}^j = \psi_{49}(X_{23}^j(3), x_9^*)\psi_{38}(X_{23}^j(2), x_8^*)\psi_{27}(X_{23}^j(1), x_7^*)w_{12}^j$$
$$-\textit{resample and synchronize}$$
$$x_{23}^j = X_{23}^j(2:3) = [X_{23}^j(2), X_{23}^j(3)], w_{23}^j = W_{23}^j = 1/N_C \quad (2.66)$$

2.2.6.3 Computing Beliefs

To estimate beliefs of unknown nodes, we compute beliefs of cliques using already computed particles from the messages. According to (2.49) - (2.51), beliefs M_1, M_2 and M_3 are function of $\psi_{C_1} m_{21}$, $\psi_{C_2} m_{12} m_{32}$ and $\psi_{C_3} m_{23}$, respectively, so we use these products as importance densities by drawing particles from them and then re-weight by remainder of (2.49) - (2.51). Let us start with M_1 and its corresponding product $\psi_{C_1} m_{21}$. As we can see in (2.40), ψ_{C_1} includes information about the distance between nodes 1 and 2, as well as between nodes 1 and 3. Moreover, message m_{21} includes information about positions of nodes 2 and 3. So we just need to locate node 1 using available positions and distances. It could be done geometrically by intersecting circles, but we prefer statistical approach which is faster. It's done by the pseudocode in Alg. 2.2 ($\text{abs}(X_{1,1}^j, X_{1,3}^j)$ is the estimated distance between these two particles, and ε is a predefined tolerance).

Algorithm 2.2: Compute belief of node 1.

1: **for** $j = 1 : N_C$ **do**
2: $X_{1,2}^j = x_{21}^j(1)$; $X_{1,3}^j = x_{21}^j(2)$;
3: **while** $(k < k_{\max})$ **do**
4: $X_{1,1}^j = \text{shift}(X_{1,2}^j, d_{12})$; $k = k + 1$;
5: **if** $\text{abs}(X_{1,1}^j,\ X_{1,3}^j) \in (d_{13} - \varepsilon, d_{13} + \varepsilon)$ **then**
6: **break;**
7: **end if**
8: **end while**
9: $W_1^j = \psi_{16}(X_{1,1}^j, x_6^*)\psi_{27}(X_{1,2}^j, x_7^*)\psi_{38}(X_{1,3}^j, x_8^*)w_{21}^j$;
10: **end for**

If we do not obtain a "good particle" after $k_{\max}$ iterations, that means that these two circles cannot intersect, so our particle is the position shifted for d_{12} in a random direction. This is not a problem because this wrong particle will obviously have later a very small weight (filtered by potential functions from anchors). The other problem is bimodality, if the circles intersect in two points; but the wrong particle will be also filtered in the same way. The same procedure is done for M_3.

As we already mentioned, the belief M_2 is not necessary since the other two cliques include positions of all unknown nodes. Anyway, we will show the procedure because it's slightly different. We have to draw particles from the product of two 4D messages and as result we expect 6D message. So, if we want to avoid to draw randomly missing particles (e.g. for message $m_{12}(x_2, x_3)$, we would have to draw single-node particles from x_4), we will directly draw particles from the product $\psi_{C_2}(x_2, x_3, x_4) m_{12}(x_2, x_3) m_{32}(x_3, x_4)$ and then re-weight by the remainder of (2.50). The following procedure shows it:

$$
\begin{aligned}
X_2^{j,1} &= [\text{shift}(x_{32}^j(2), d_{24}), x_{32}^j(1), x_{32}^j(2)] \\
X_2^{j,2} &= [x_{12}^j(1), x_{12}^j(2), \text{shift}(x_{12}^j(1), d_{24})] \\
X_2^j &= \text{choose}(X_2^{j,1} \cup X_2^{j,2}, N_C); W_2^j = 1/N_C \\
W_2^j &= \frac{m_{12}(X_{2,2}^j, X_{2,3}^j) m_{32}(X_{2,3}^j, X_{2,4}^j)}{m_{12}(X_{2,2}^j, X_{2,3}^j) + m_{32}(X_{2,3}^j, X_{2,4}^j)} W_2^j \\
W_2^j &= \psi_{27}(X_{2,2}^j, x_7^*) \psi_{38}(X_{2,3}^j, x_8^*) \psi_{49}(X_{2,4}^j, x_9^*) W_2^j
\end{aligned} \tag{2.67}
$$

where function choose$(X_2^{j,1} \cup X_2^{j,2}, N_C)$ chooses randomly N_C particles from $2N_C$. Also, for simplicity, through the delivery, updated and old particles are denoted by the same symbols. This procedure is known as *importance sampling* [D22], the approximation of original distribution ($m_{12}m_{32}$) with proposal one ($m_{12} + m_{32}$) from which is easy to draw samples ($X_2^{j,1} \cup X_2^{j,2}$), and then re-weighting ($m_{12}m_{32}/(m_{12} + m_{32})$) to compensate the error.

The final estimates of unknown nodes are given by the mean values of particles from the clique C_m :

$$
x_u^{est} = \sum_{j=1}^{N_C} W_m^j X_{m,u}^j / \sum_{j=1}^{N_C} W_m^j \tag{2.68}
$$

2.2.6.4 Improved Sampling Procedure

There is an important modification to this algorithm that can reduce significantly the initial number of particles. As we already mentioned, if we draw *N* particles from one node, generally it corresponds to $N_C = N^3$ particles of a 3-nodes clique. However, we included information about distance, so our new number for the same clique is $N_C = NN_\theta^2$ where N_θ represents the number of possible angles. But this number is still very large, so we would like to include additional information.

We assumed that there is no *a priori* information about node position. However, after the very first phase of the algorithm, we computed joint messages M_{12} and M_{32} which include current information about positions of the cliques C_1 and C_3 . At this point, particles are concentrated in a smaller region (except for very few of them), so we can draw a new set of particles around single-node particles of the joint messages. For C_1 , it's done by the following procedure:

$$
\begin{aligned}
d^j = \text{Unif}(0, r), X_{1,1}^j &= \text{shift}(X_{1,1}^j, d^j) \\
X_{1,2}^j = \text{shift}(X_{1,1}^j, d_{12}), X_{1,3}^j &= \text{shift}(X_{1,1}^j, d_{13}) \\
W_1^j &= 1/N_C \\
-\textit{compute messages again}&
\end{aligned} \tag{2.69}
$$

where *r* is the radius of deployment area of new particles. Computing messages again is mandatory since we draw new set of particles, which means that we have to run the algorithm from the beginning. Of course, for C_3 , we use the same procedure. This improved procedure allows us to decrease initial number of samples to $N_C = NN_\theta^2/n$ where *n* is the *reducing factor* that could be found experimentally (by establishing the same accuracy in the old and new case and computing N_C^{old}/N_C^{new}). It's also proportional to the ratio of the new to the old deployment area. The new set of particles from C_1 is shown in Figure 2.16.

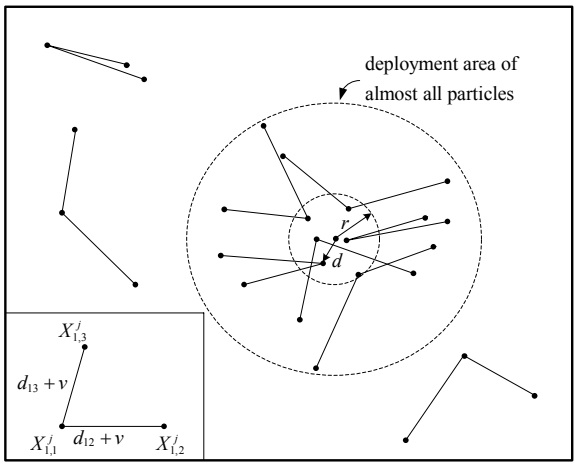


Figure 2.16: Improved set of particles from clique C_1. The particles are concentrated in a smaller region (except for very few of them).

2.2.6.5 Simulation Results

We simulated the network from Figure 2.13 using NBP, GBP-JT and NGBP-JT algorithms. We placed 10 nodes in $2\text{m}\times2\text{m}$ area, 5 anchors and 5 unknowns. We set the values of transmission radius ($R = 25\%$ of diagonal length of the deployment area) and standard deviation of measured distance ($sigma = 0.1\text{m} = 14\%R$). The number of iterations for NBP is set to the length of the longest path in the graph ($N_{iter} = 5$), and for GBP-JT/NGBP-JT there is obviously, in our case, just one iteration. The number of particles for NBP is set to $N = 400$, so the corresponding number of grid points for GBP-JT is $N_G = 20\times20$. For NGBP-JT we used improved sampling procedure with radius $r = 0.1$ and experimentally we found out the reducing factor which does not change the accuracy ($n \approx 4$). Assuming that minimum number of angles could be approximated with $N_\theta = 10$, we set the number of clique's particles to $N_C = NN_\theta^2/n = 10000$.

We ran the simulation for NBP and NGBP-JT, and obtained results are shown in Figure 2.17. Obviously, the location estimates for the NGBP-JT are more accurate since this algorithm is correct for network with loops. NBP algorithm does not converge well for a few nodes, but for some other values of parameters, or with different positions of some nodes, it provides estimates with almost same accuracy as NGBP-JT. However, comparing uncertainties for NBP and NGBP-JT (contours in Figure 2.17), we can see that NBP provides us better guarantees of its estimate. This is because NBP algorithm, at each iteration, computes relative direction of samples using information from previous iteration [Ihler05].

Moreover, we checked the averaged accuracy with respect to the deviation of measured distance for all three methods (Figure 2.18a). The accuracy of GBP-JT is always higher than accuracy of NBP and NGBP-JT. NGBP-JT provides us better accuracy than NBP for some usual values of distance deviation (e.g. for measurements using *time of arrival*, the error is 5-20 $\%R$ [PAK+05]), and unexpectedly worse accuracy for higher values of the mentioned deviation. Anyway, this accuracy could be increased, using larger number of particles (e.g. increasing N_θ), until the "bottom line" defined by the accuracy of GBP-JT.

Comparing with NBP/NGBP-JT, the computational cost of GBP-JT is, of course, very large (62 MFlops) and absolutely unacceptable. Nonparametric approximation of this algorithm decreased it around 25 times, and improved sampling additional 4 times. So the final computational cost of NGBP-JT in simulated example is 1.58 MFlops, around double as many comparing with NBP (0.67 MFlops). More general result, with respect to the number of single-node particles, is given in Figure 2.18b (for NGBP-JT it's additionally necessary to multiply x-axis by $N_C/N = 25$). Obviously, we didn't plot results for GBP-JT since it's significantly above the axis.

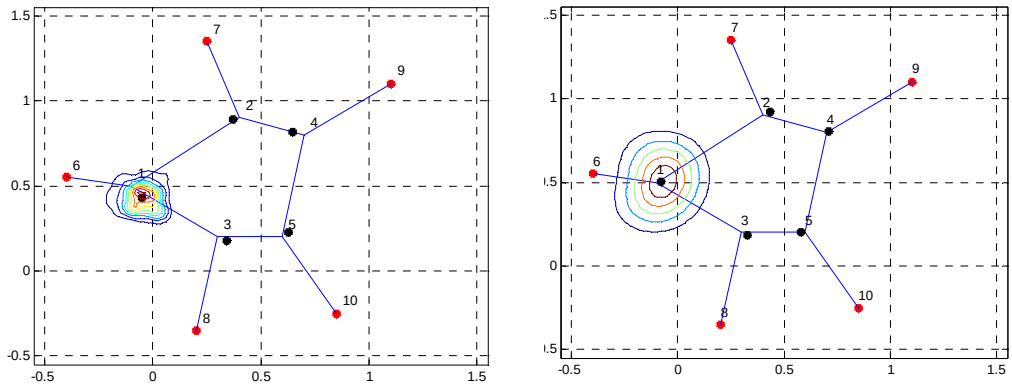


Figure 2.17: Comparison of the results for a 10-node network (a) NBP, (b) NGBP-JT.

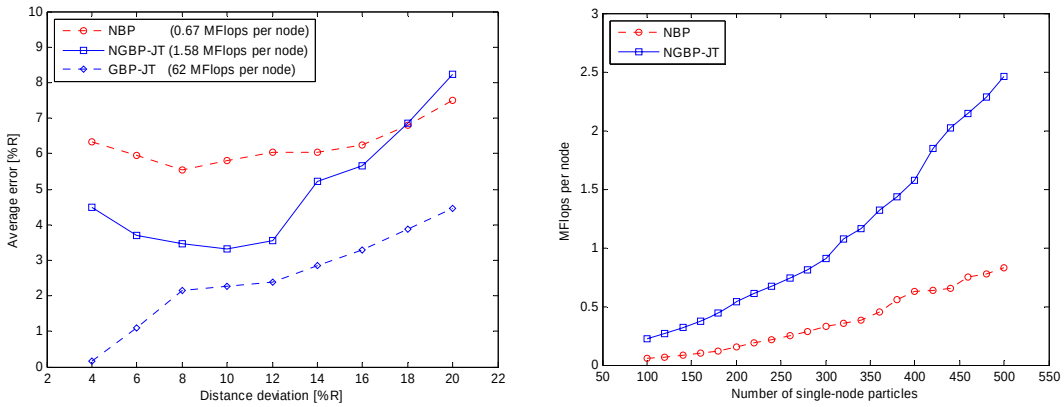


Figure 2.18: Comparison of (a) accuracy and (b) computational cost.

2.2.7 Nonparametric Belief Propagation Based on Spanning Trees (NBP-ST)

The previous methods are still very complex for the large-scale ad hoc sensor networks. Moreover, the connectivity in these networks is very high which makes computational and communication burdens for low-power applications. Therefore, we will implement NBP based on spanning trees (NBP-ST) created by breadth first search (BFS) method [BM06]. There are more similar methods, so we refer the reader to the one [WJW01, WJW03] which is not based on standard BP.

2.2.7.1 Spanning Tree Formation

We start by describing the basics of graphical models. An undirected graph $G = (V, E)$ consists of a set of nodes or vertices V that are joined by a set of edges E. A *loop* or *cycle* is a sequence of distinct edges forming a path from a node back to itself. A *tree* is a connected graph without any loops. A *spanning tree* is an acyclic subgraph that connects all the nodes of the original graph. A *root node* is a node without parent and *leaf node* is a node without children. In order to define an undirected graphical model, we place at each node a random variable x_s taking values in some space. In case of localization, this random variable represents the 2D position and each edge represents the measured distance. If we exclude anchor nodes, the graph is obviously undirected but only for the first phase (spanning tree formation) we assume that it is directed (starting from chosen root node).

The optimal method for spanning tree formation for unweighted graphs is *breadth first search* (BFS). It begins at the root node and explores all the neighboring nodes. Then each of those neighbors explores their unexplored neighbor nodes, and so on, until all nodes are explored. In this way, there will not be

loops in the graph because all nodes will be explored just once. The detailed pseudocode is shown in Alg. 2.3. The worst case complexity is $O(v+e)$, where v is the number of nodes and e is the number of edges in the graph, since every node and every edge will be explored in the worst case.

Algorithm 2.3: Breadth First Search (BFS).

```
Input: list of nodes Q and root node root
Set current root: r ← root
while Q is not empty do
    for all nodes t ∈ G_r do
        if t ∈ Q then
            Remove t from Q
            Insert t in Q_r
            Insert d_rt in S
        end if
    end for
    Set current root: r ← first unused node from Q_r
end while
Output: spanning tree {Q, S}
```

In case of NBP localization, we exclude all the anchors from the BFS algorithm since they do not form the loops in the graph (they just send, and never receive the messages). A graph generally has a large number of spanning trees, but since our graph is unweighted we choose few (minimum 2) of them in a partly random way. The first root node is chosen randomly from the set of all unknown nodes. In order to maximize the difference between two spanning trees, the second root node has to be as far as possible from the first root node. Thus, it should be one of the leaf nodes. If we want to form more spanning trees, the analog constraint will be used. An example of a loopy graph and two corresponding spanning trees, formed by BFS with mentioned constraints, are illustrated in Figure 2.19. Note that, using BFS, it is not possible to form two spanning trees with completely different edges and that always some edges will be out of both spanning trees. If we want to include all edges, we have to add more spanning trees but it is usually not necessary since it will only provide us redundant information. It is especially the case in the networks with high connectivity.

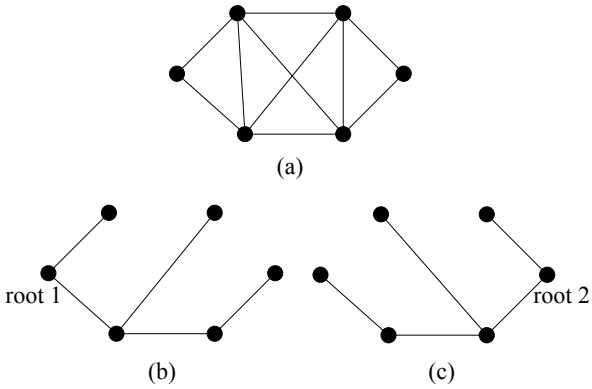


Figure 2.19: (a) Example of loopy graph, (b), (c) Spanning trees created by BFS method.

The NBP method is naturally distributed through the graph which means that there is no central unit which will handle all computations. Therefore, the proposed BFS method has to be done in a distributed way. This can be simply done if each unknown node initially broadcast its *ID* to all neighbors, which will continue to broadcast to others, and so on, until each unknown node has a list of all unknown nodes in the graph. One node (e.g. with lowest *ID*) has to be assigned to choose the root node from the list and give it a permission (by multihop broadcasting) to start BFS algorithm. Then, the chosen root node has all initial data to start BFS algorithm, and, when it's necessary, has only to broadcast all data (i.e. variables from Alg. 2.3) to all its neighbors.

Finally, NBP-ST algorithm represents two (or more) independent runnings of the NBP algorithm based on formed spanning trees. Each running will provide us weighted particles of the node beliefs computed by standard NBP. The simplest way to fuse these beliefs is to *resample with replacement* [AMG+02] from weighted particles from all spanning trees, which produces the particles with same weights. Then, the final location estimate is just the mean value of the particles from all spanning trees. The pseudocode in Algorithm 2.4 illustrates NBP-ST method.

Algorithm 2.4: NBP-ST method for localization.

1: **for all** nodes **do**
2: Take sensing actions
3: Set all parameters to the initial values
4: Broadcast own and all received *IDs* and listen for other sensor broadcasts (until receive all *IDs*)
5: **end for**
6: Set a list of nodes for BFS (excluding anchors): Q
7: Choose randomly root node from the list Q: *root*
8: **for all** *spanning trees* **do**
9: Run BFS (Alg. 3)
10: Run NBP on defined spanning tree
11: Choose root node as far as possible from the previous roots
12: **end for**
13: Fuse all beliefs into one and compute location estimates

2.2.7.2 Simulation Results

In the simulations, we placed 100 unknown and 10 anchor nodes randomly in 20m x 20m area. Since the unknown nodes near the edges of deployment area suffer from low connectivity, we include one realistic constraint: four anchors are randomly deployed within four areas 4m x 4m near the edges, respectively. The standard deviation of the Gaussian noise is set to *sigma* = 0.3 m and the number of iteration is set to $N_{iter} = 3$. All simulations are done for $N = 50$ and $N = 100$ particles with respect to the transmission radius ($R = 4$ m - 10m). The error is defined as a distance between true and estimated location. To measure the communication cost, we count *elementary messages*, where one elementary message is defined as one coordinate of one particle. Finally, each point in the simulations represents the average over 20 Monte Carlo trials.

Using the defined scenario, we compared NBP and NBP-ST algorithms. For NBP-ST, we used 2 spanning trees. The original network and 2 spanning trees created by BFS are illustrated in Figure 2.20 and Figure 2.21, respectively. Regarding accuracy and coverage (percentage of located nodes with error less than predefined tolerance) in Figure 2.22, NBP-ST performs better than NBP for $R > 7$ m, approximately. Obviously, for these values of R there is a large number of loops in the network which decreases the performance of NBP method. For lower values of R, we could expect that NBP-ST performs with higher (or same) accuracy, but we cannot forget that, by using only 2 spanning trees, we didn't include all information (removed edges) that we have. Thus, the NBP overperforms NBP-ST in this case.

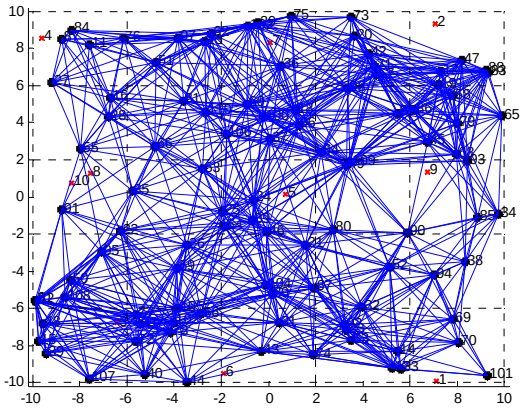
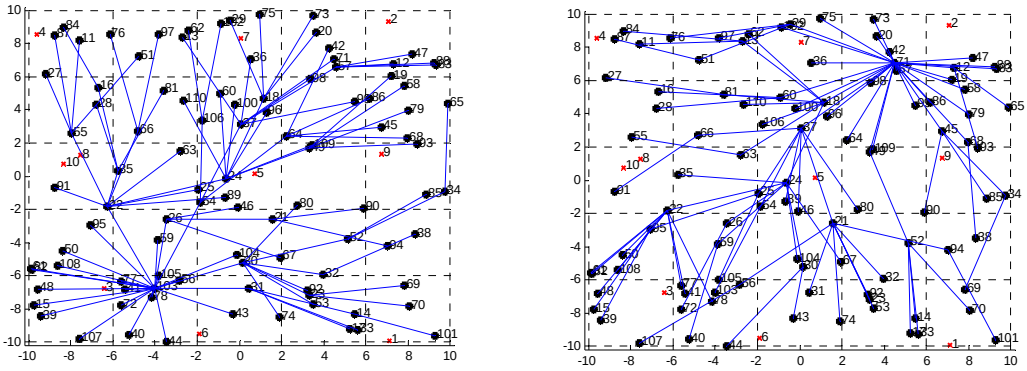


Figure 2.20: Original network with 100 unknown nodes (black dots) and 10 anchors (red asterisks) for $R = 6m$. The lines represents observed (1-step) links (the links from the anchors are not shown).

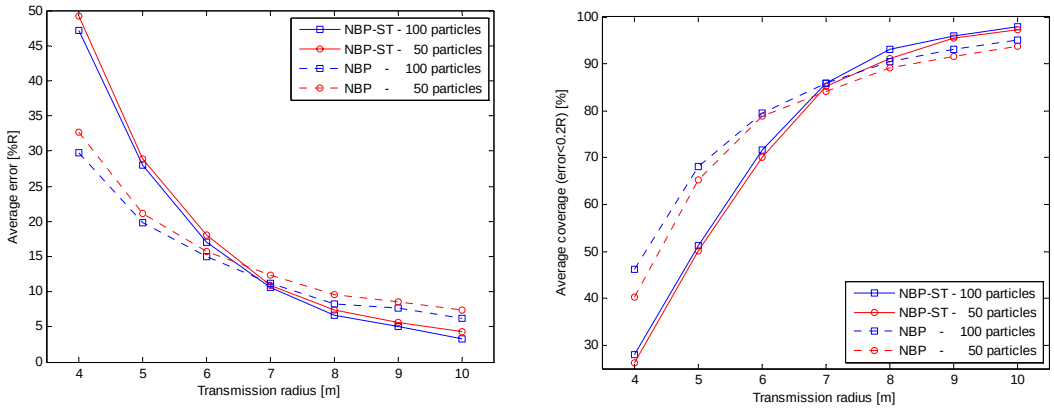


(a) (b)

Figure 2.21: Two spanning trees for the network from Figure 11. The roots are (a) node 103 and (b) node 71.

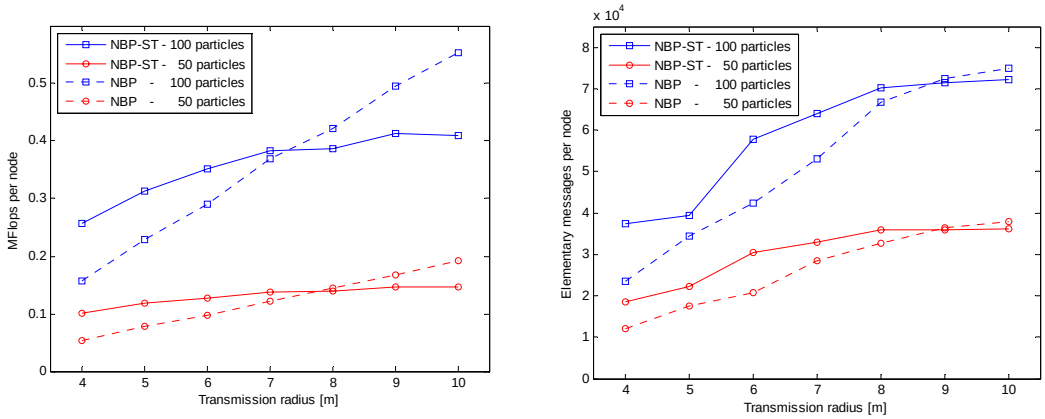
Regarding computational/communication cost (Figure 2.23), NBP-ST performs better than NBP for $R > 8$ m and $R > 9$ m, respectively. In order to explain this we have to remember two main things taken into account: removing the edges in order to form the spanning trees and running NBP two times in these spanning trees. First operation decreases the costs, but the second one increases it. Therefore, in low connected networks the second operation predominates, but in high connected networks the first one predominates. The main contribution here is that for high transmission radius, computational and communication cost are nearly constant. Of course, one could trade these costs for the accuracy by adding more spanning trees.

The final conclusion is that NBP-ST algorithm performs better than NBP in all terms, for $R > R_{min}$. In our case $R_{min} = 9$ m, but this parameter depends on the density in the network (i.e. average connectivity). These values of transmission radius are available in all today's wireless sensor nodes even if they are set to work in low-power mode (e.g. up to 75m for sensors based on ZigBee, IEEE 802.15.4 [IEEEstd06]).



(a) (b)

Figure 2.22: Comparison of (a) accuracy and (b) coverage.



(a) (b)

Figure 2.23: Comparison of (a) computational and (b) communication cost.

2.2.8 RSS/TOA Based Distance Modeling

We are going to use real database of measured RSS and TOA obtained in indoor environment (Figure 2.24), in order to to obtain distances between transmitters (M1, M4, and M8) and each of the 30 receivers (2 receivers at each point). We provide the hardware description, RSS/TOA calibration, and the error models. These results will justify importance of probabilistic methods described in previous sections since they are capable to handle non-Gaussian uncertainties.

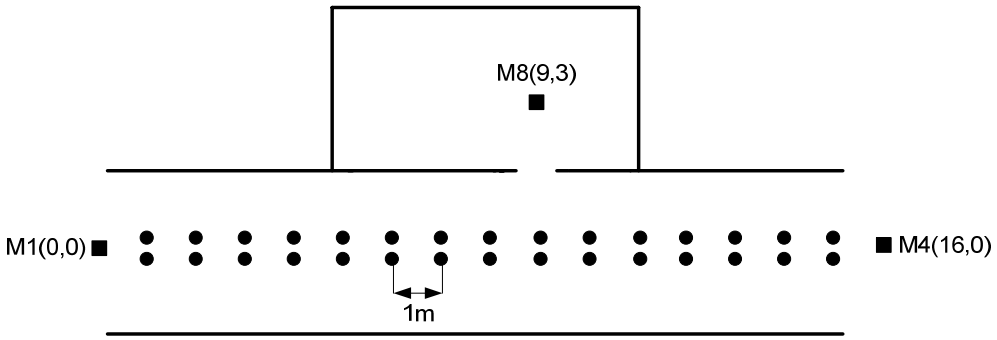


Figure 2.24: The illustration of deployment. There are 3 transmitters (marked with black boxes) and 30 receivers (marked with black circle).

2.2.8.1 Hardware Description

The SDR platform developed at UPM is based on the products provided by Sundance. It is based on one full-length PCI carrier board (SMT310Q) which provides access to four, industry standards, Texas Instruments Modules (TIMs) format. The SDR platform used in WHERE project includes C64xx-based module (SMT365) combined with a dual high-speed ADC/DAC module (SMT370), both plugged on a SMT310Q carrier board. In addition, it also includes IF/RF front-end module (SMT349) which is delivered together with two omni-directional antennas. The antennas work in a range of 2.4GHz to 2.5GHz while supporting bandwidths of up to 20MHz. The main characteristics of SDR platform include:

- Two 14-bit ADCs (AD6645-105) sampling at up to 105MHz, DC coupled
- Dual 16-bit DAC (AD9777) sampling at up to 400MHz (interpolation)
- IF output signal is 70MHz ± 8MHz,
- The 70MHz IF is converted to a 2nd IF of 374 MHz
- RF output signal is in the 2.4–2.5 GHz ISM band,
- Transmitter gain control: 31dB, in 1dB steps

- Transmitter gain accuracy: ±2dB maximum
- RF bandwidth: 20MHz
- Signal type: WiFi
- Transmit power: 10mW

In order to obtain the correct estimation of the time of arrival it is essential that AD converter at receiver side, and DA converter at transmitter side start at exactly the same time. This is achieved with the introduction of a 60MHz oscillator which delivers the reference clock to the receiver and transmitter. It is located on a separate logical board together with a simple D flip-flop chip. The purpose of the chip is to synchronize the trigger, which is generated by general I/O pin at receiver, with the reference clock. In this way the trigger reaches transmitter and receiver always in same conditions with respect to sampling frequency. With these modifications we achieved robust time reference but the cost is presented in form of extra cabling which complicates the system and reduces its mobility.
For the TOA estimation we have implemented simple approach based on the location of the first peak in time domain. As previously explained, the sampling frequency is 60MHz and the consistency between time references of transmitter and receiver is achieved by an external trigger. However, in order to achieve better precision the received signal is interpolated with factor of 20. In this way each interpolated sample is separated 0.83ns which theoretically corresponds to 0.25m of path difference.

2.2.8.2 RSS/TOA Calibration

Using obtained measurements, we need to estimate the distance between each pair of transmitters and receivers. For RSS measurements, we use log-normal model (RSS=RSS0-10·np·log(d/1m)), and for TOA linear model (d=v·TOA+d0), as defined in following tables.. Since for each of the 3 transmitters, there are two receivers at same distance, we will make 6 different models. Furthermore, we compare these models with realistic models obtained by linear interpolation. These parameters are shown in Table 2.3 and Table 2.4, and obtained models in Figure 2.25 to Figure 2.30.

Table 2.3: Parameters for RSS log-normal model.

	path-loss exponent (np)	reference power (RSS0) [dBm]
Transmitter M1 - Receivers 1 (LOS)	1.47	-23.81
Transmitter M1 - Receivers 2 (LOS)	1.86	-18.76
Transmitter M4 - Receivers 1 (LOS)	1.57	-13.73
Transmitter M4 - Receivers 2 (LOS)	0.86	-19.35
Transmitter M8 - Receivers 1 (NLOS)	3.7	-10.55
Transmitter M8 - Receivers 2 (NLOS)	3.46	-13.30

Table 2.4: Parameters for TOA linear model.

	speed (v) [m/s ·10^08]	offset (d0) [m]
Transmitter M1 - Receivers 1 (LOS)	1.776	-2.79
Transmitter M1 - Receivers 2 (LOS)	1.384	-0.45
Transmitter M4 - Receivers 1 (LOS)	2.059	-5.39
Transmitter M4 - Receivers 2 (LOS)	2.394	-6.88
Transmitter M8 - Receivers 1 (NLOS)	0.814	0.25
Transmitter M8 - Receivers 2 (NLOS)	0.492	1.70

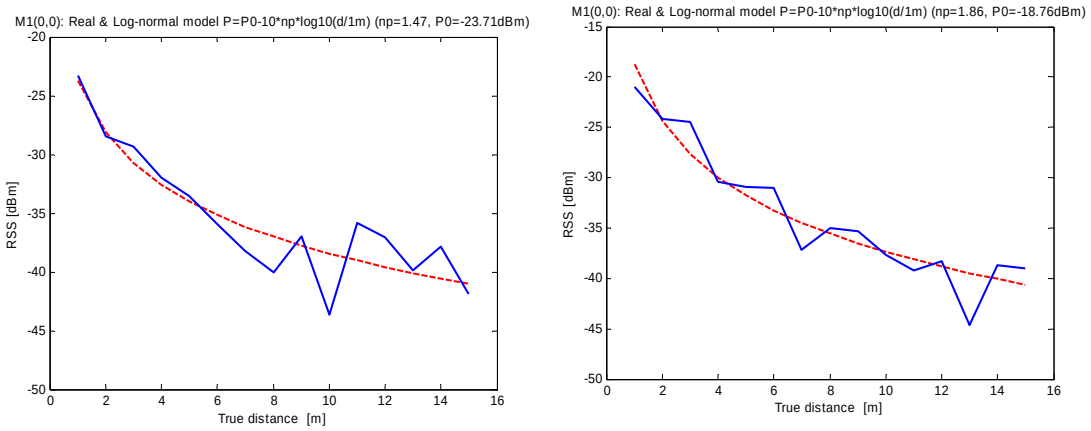


Figure 2.25: Real vs lognormal calibration for transmitter M1(0,0) and receivers (a) 1, (b) 2.

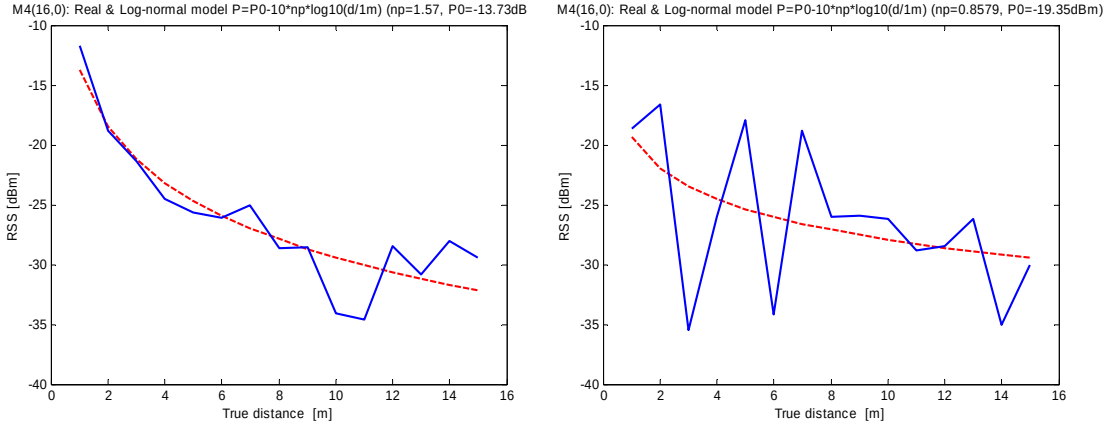


Figure 2.26: Real vs lognormal calibration for transmitter M4(16,0) and receivers (a) 1, (b) 2.

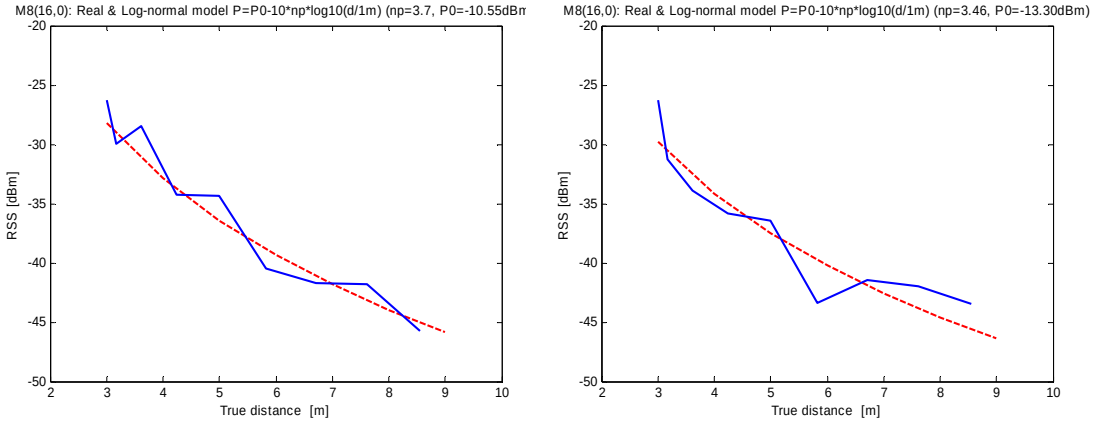


Figure 2.27: Real vs lognormal calibration between transmitter M8(9,3) and receivers (a) 1, (b) 2.

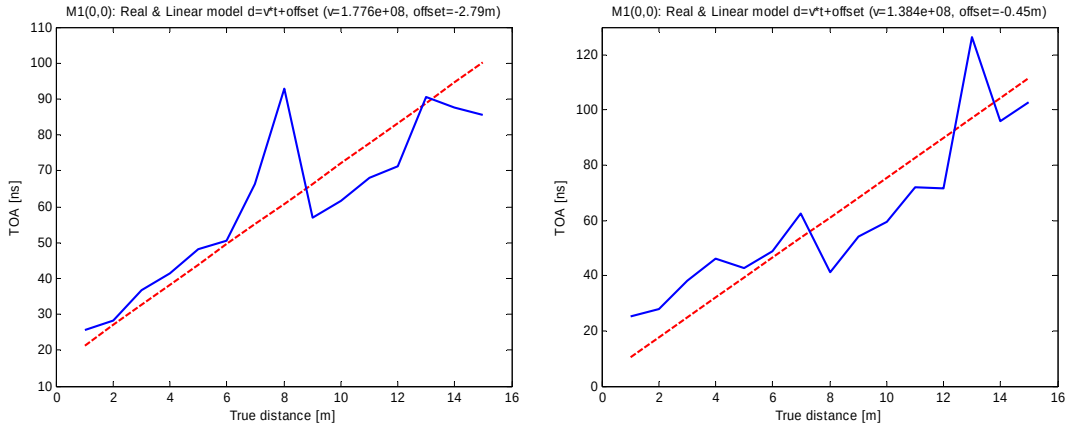


Figure 2.28: Real vs linear calibration between transmitter M1(0,0) and receivers (a) 1, (b) 2.

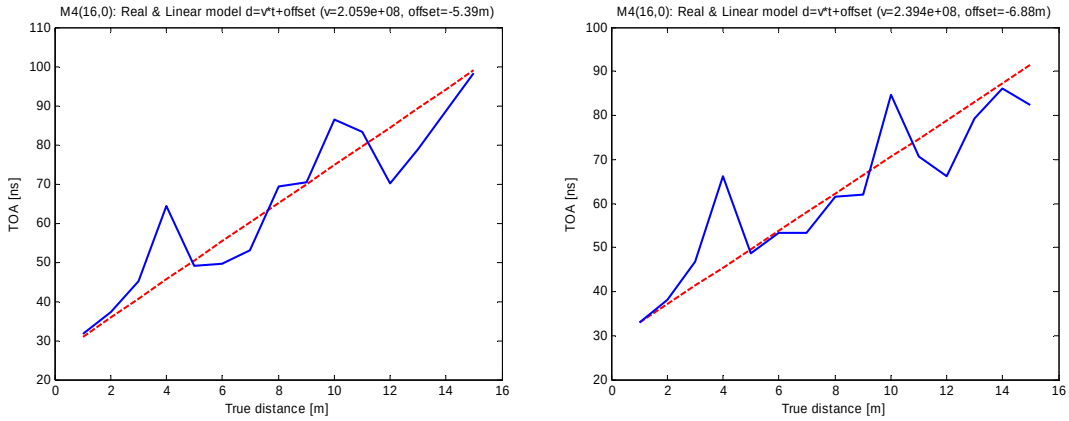


Figure 2.29: Real vs linear calibration between transmitter M4(16,0) and receivers (a) 1, (b) 2.

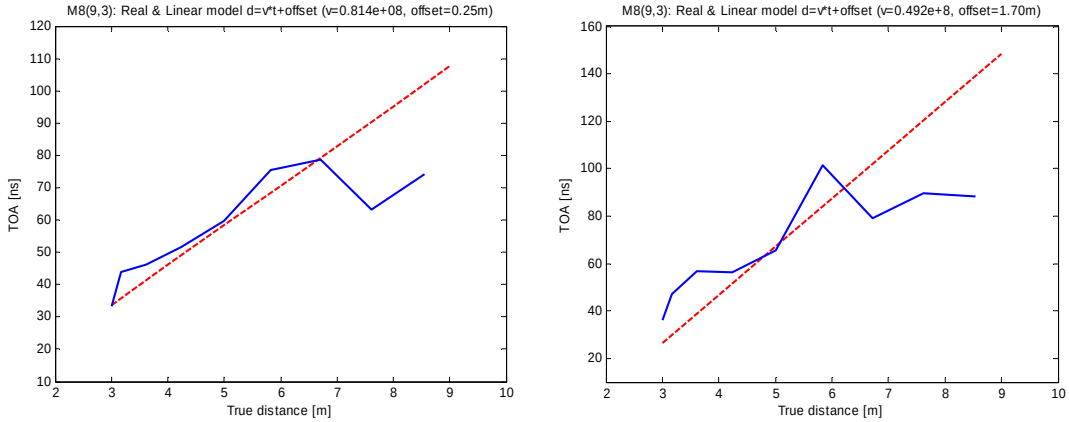


Figure 2.30: Real vs linear calibration between transmitter M8(9,3) and receivers (a) 1, (b) 2.

2.2.8.3 Gaussian Mixture Error Models

In contrast to previous section, we merged both receivers which are placed at the same distance from transmitter. In Figure 2.31 to Figure 2.36, we show the histograms of the errors and appropriate Gaussian mixture fitting. We choose 3 mixtures of Gaussian with following parameters (found by *k-means* clustering):

Table 2.5: Gaussian Mixture Parameters (RSS).

	(weight1, mean1, var1)	(weight2, mean2, var2)	(weight3, mean3, var3)
Transmitter M1 (LOS)	(0.33, -1.83, 2.76)	(0.46, 0.02, 0.33)	(0.21, 3.01, 22.20)
Transmitter M4 (LOS)	(0.35, -2.97, 5.52)	(0.48, 0.56, 1.39)	(0.17, 13.45, 577.38)
Transmitter M8 (NLOS)	(0.35, -1.01, 0.17)	(0.33, 0.04, 0.12)	(0.32, 1.30, 0.48)

Table 2.6: Gaussian Mixture Parameters (TOA).

	(weight1, mean1, var1)	(weight2, mean2, var2)	(weight3, mean3, var3)
Transmitter M1 (LOS)	(0.43, -1.91, 0.40)	(0.29, 0.24, 0.18)	(0.28, 1.67, 3.20)
Transmitter M4 (LOS)	(0.33, -1.45, 1.19)	(0.43, -0.11, 0.11)	(0.24, 1.57, 3.43)
Transmitter M8 (NLOS)	(0.26, -1.24, 1.02)	(0.37, 0.04, 0.13)	(0.37, 1.20, 0.38)

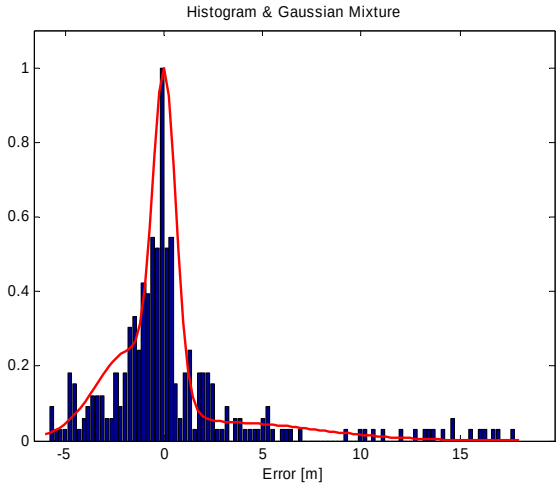


Figure 2.31: Gaussian mixture error model for distance obtained from RSS (transmitter M1(0,0), both receivers, LOS).

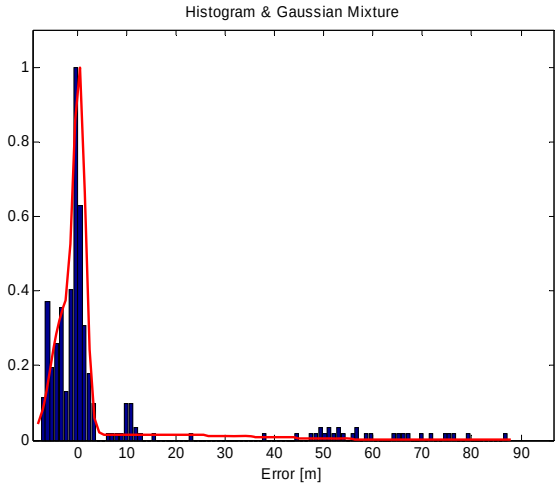


Figure 2.32: Gaussian mixture error model for distance obtained from RSS (transmitter M4(16,0), both receivers, LOS).

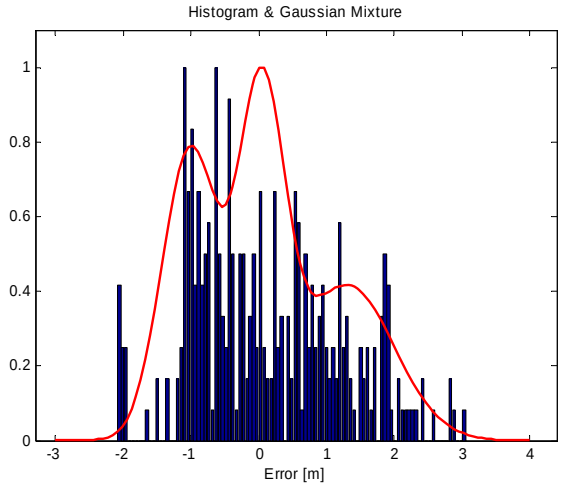


Figure 2.33: Gaussian mixture error model for distance obtained from RSS (transmitter M8(9,3), both receivers, NLOS).

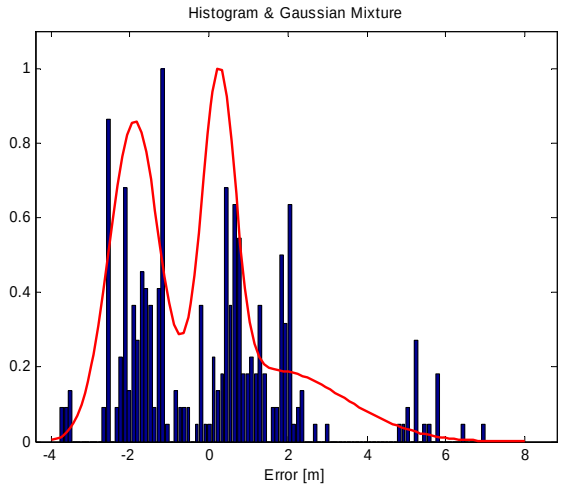


Figure 2.34: Gaussian mixture error model for distance obtained from TOA (transmitter M1(0,0), both receivers, LOS).

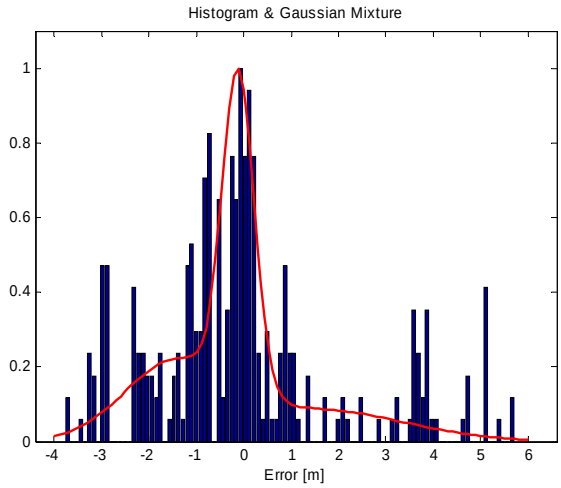


Figure 2.35: Gaussian mixture error model for distance obtained from TOA (transmitter M4(16,0), both receivers, LOS).

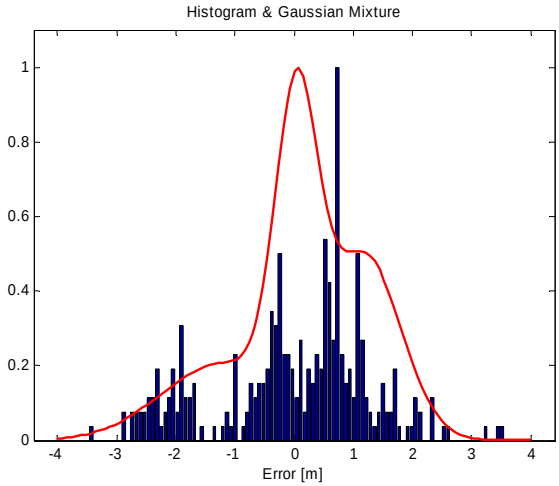


Figure 2.36: Gaussian mixture error model for distance obtained from TOA (transmitter M8(9,3), both receivers, NLOS).

2.2.8.4 Distance-Dependent Gaussian Error Models

The parameters of previous models do not depend on distance. This is usually not true, especially in indoor scenario, so now we are going to make distance-dependent model. Since we do not have sufficient data to make Gaussian mixture for each distance, we are going to fit the data to Gaussian model and find appropriate mean values and standard deviations. We fit these parameters using polynomial (3-degree) least-square regression ($\mu = A3 \cdot d^3 + A2 \cdot d^2 + A1 \cdot d + A0$, $\sigma = B3 \cdot d^3 + B2 \cdot d^2 + B1 \cdot d + B0$). These parameters are shown in Table 2.7 and Table 2.8, and comparison between obtained dependencies (real and polynomial) in Figure 2.37 to Figure 2.42.

Table 2.7: Mean value (μ) coefficients.

μ	M1 RSS	M4 RSS	M8 RSS	M1 TOA	M4 TOA	M8 TOA
A3	-0.00716	-0.0287	0.15258	0.003932	0.003993	-0.15058
A2	0.13038	0.567937	-2.89436	-0.09059	-0.10878	2.664325
A1	-0.44192	-2.75445	17.81386	0.349219	0.66612	-15.2351
A0	0.183468	3.088489	-34.5727	0.920237	-0.17013	27.5333

Table 2.8: Standard deviation (σ) coefficients.

σ	M1 RSS	M4 RSS	M8 RSS	M1 TOA	M4 TOA	M8 TOA
B3	-0.00885	-0.00261	0.180981	-0.00038	-0.00231	-0.09864
B2	0.189434	0.045162	-3.2325	0.00024	0.044887	1.658973
B1	-0.73822	-0.10685	18.10897	0.088794	-0.1393	-8.70079
B0	1.023988	0.181454	-30.4695	0.504382	0.391983	14.78034

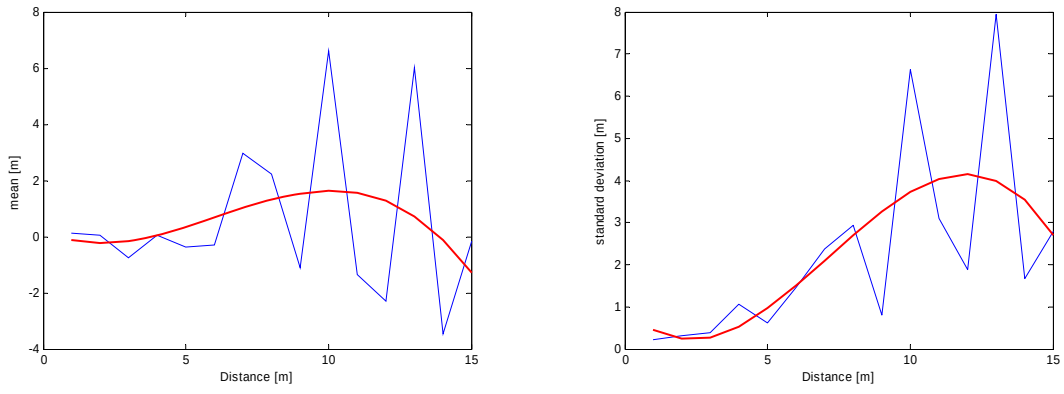


Figure 2.37: Distance-dependent RSS Gaussian error model: (a) mean, (b) standard deviation (transmitter M1(0,0), both receivers, LOS).

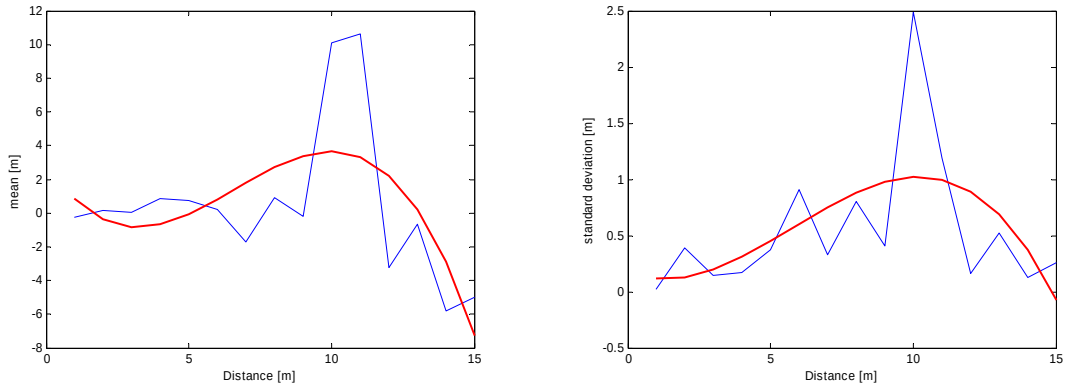


Figure 2.38: Distance-dependent RSS Gaussian error model: (a) mean, (b) standard deviation (transmitter M4(16,0), both receivers, LOS).

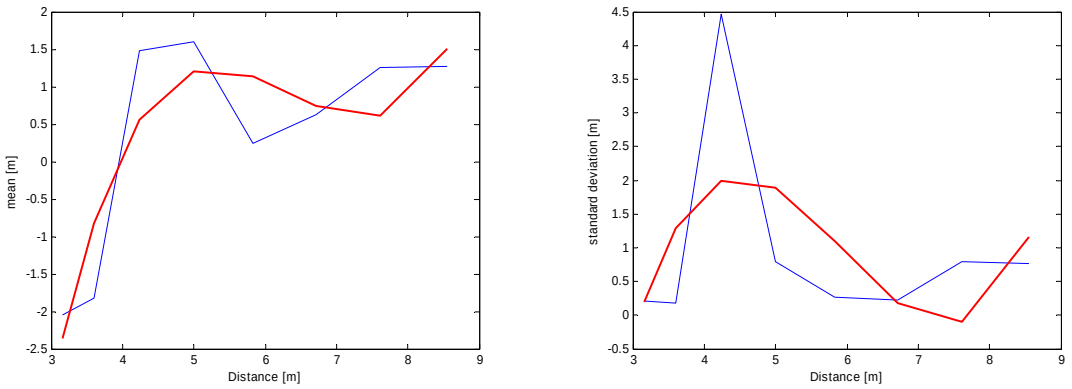


Figure 2.39: Distance-dependent RSS Gaussian error model: (a) mean, (b) standard deviation (transmitter M8(9,3), both receivers, NLOS).

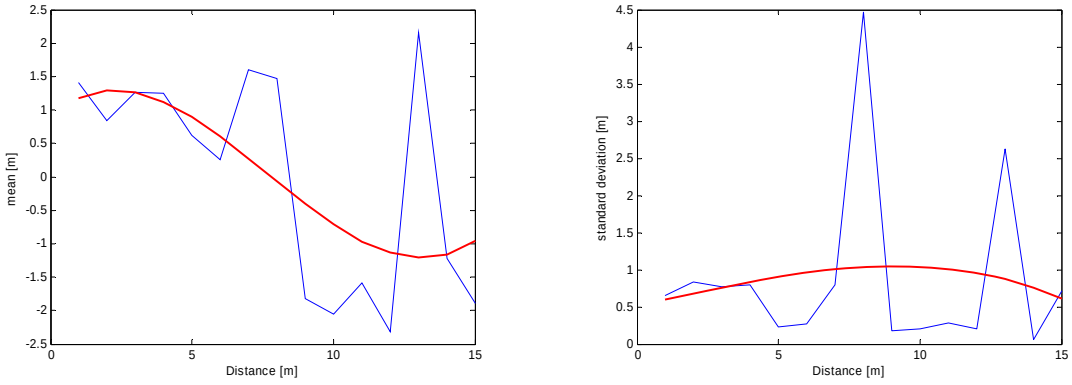


Figure 2.40: Distance-dependent TOA Gaussian error model: (a) mean, (b) standard deviation (transmitter M1(0,0), both receivers, LOS).

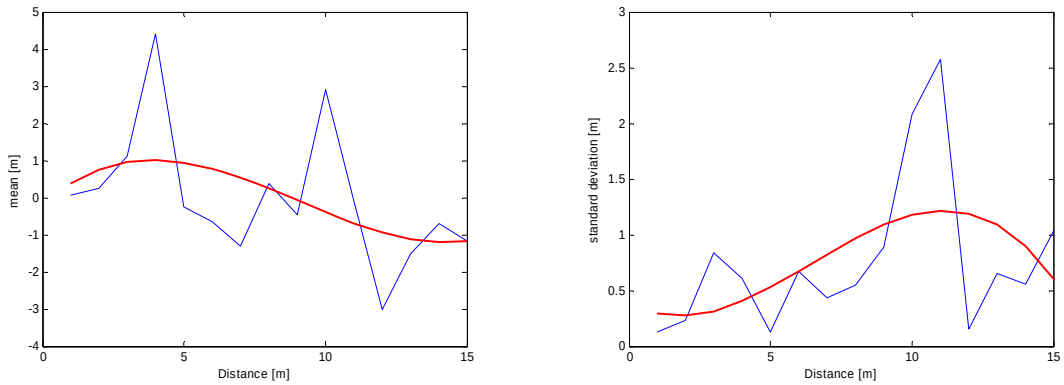


Figure 2.41: Distance-dependent TOA Gaussian error model: (a) mean, (b) standard deviation (transmitter M4(16,0), both receivers, LOS).

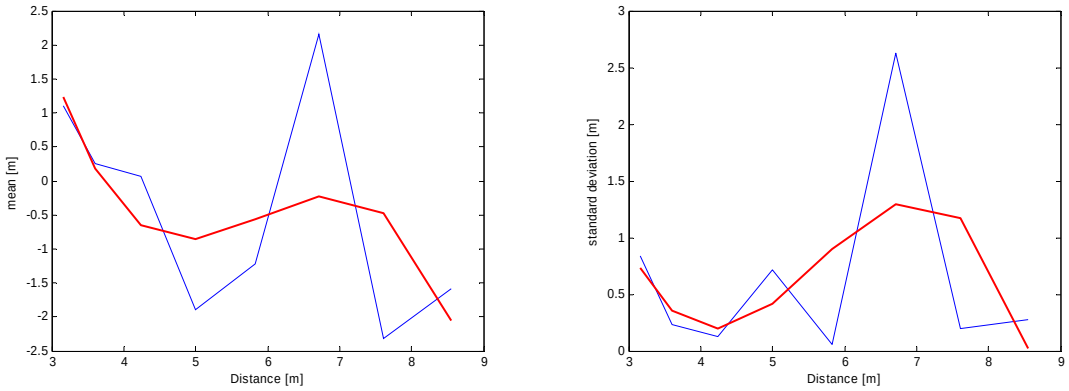


Figure 2.42: Distance-dependent TOA Gaussian error model: (a) mean, (b) standard deviation (transmitter M8(9,3), both receivers, NLOS).

2.2.8.5 Conclusion

We used real database of measured RSS and TOA obtained in indoor environment. We first obtained (by calibration) appropriate parameters for RSS log-normal and TOA linear models and then we obtained 2 error models: Gaussian mixture distance-independent model, and Gaussian distance-dependent model. We can conclude that in indoor scenario, the Gaussian model is not appropriate, and that even Gaussian mixture is not capable to fit all uncertainties. On the other hand, distance-dependent Gaussian (mixture) model is capable to handle larger scope of uncertainties. In addition, distance-dependent models for RSS-LOS scenario are close to theoretical model since mean and variance are increasing with distance (however, it's valid only up to 10m, probably due to the outliers). Finally, the obtained models are not very convenient (especially in NLOS scenario) because they can be different even in same environment. To avoid this, we need to have better statistics of the measured data (e.g., by repeating the same measurement campaign several days).

2.2.9 Conclusion and Future Work

In this section, we reviewed probabilistic localization technique based on belief propagation (BP) and nonparametric (particle based) approximation of BP (NBP). Since BP/NBP method have a problem in loopy networks, we propose four methods: generalized BP based on Kikuchi approximation (GBP-K), generalized BP based on junction-tree method (GBP-JT), nonparametric GBP-JT (NGBP-JT) and NBP based on spanning trees (NBP-ST). We show that the last one (NBP-ST) is currently the unique method which is computationally feasible in large-scale ad-hoc/sensor networks. In addition, we use real database in order to obtain more realistic model for indoor scenario. The obtained models justify importance of all probabilistic methods since they are capable to handle non-Gaussian uncertainties.

There remain many open directions for the future work. The most important is the generalizing NGBP-JT method for large-scale ad-hoc/sensor networks using some efficient method for formation of junction tree cliques within the network. Moreover, including RSS/TOA indoor data, in some of the described

algorithms, will provide us more precise conclusions about the performance. Finally, real-time target tracking using these methods could be an interesting direction.

2.3 A Mean Field Algorithm for Sensor Self-Localization in Cooperative Wireless Networks

The problem of sensor self-localization in cooperative networks requires an algorithm for position estimation. A standard solution approach to this problem is to apply iterative methods based on belief propagation (BP) [IFMW05], [WLW09]. An alternative to the BP algorithm is the mean field algorithm [Minka2005]. In the sequel, we develop a localization algorithm for cooperative wireless networks based on mean field theory. In a highly simplified simulation scenario, we compare the performance of the derived algorithm to the maximum a posteriori (MAP) estimator performance and the Cramer-Rao lower bound (CRLB).

2.3.1 Model

We consider a situation in which a set of wireless network nodes are randomly scattered in a plane region. The set of nodes is divided into *anchor* nodes with known positions, and *mobile* nodes at unknown positions. The position of node t is described by the two-dimensional vector $\mathbf{x}_t = [x_{t1}, x_{t2}]^T$, where $(\cdot)^T$ is the transpose operator. A node's prior knowledge of its position can be described by a circular symmetric Gaussian distribution with mean μ_t and variance σ_t^2

$$p_t(\mathbf{x}_t) = \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left(-\frac{\|\mathbf{x}_t - \mu_t\|^2}{2\sigma_t^2} \right), \tag{2.70}$$

where $\|\cdot\|$ denotes the Euclidean norm. The prior position knowledge may be uninformative (i.e. the variance may be large).

For model derivation, we consider only a situation similar to the one depicted in Figure 2.43 where each of the nodes have communication links to a subset of neighbouring nodes.

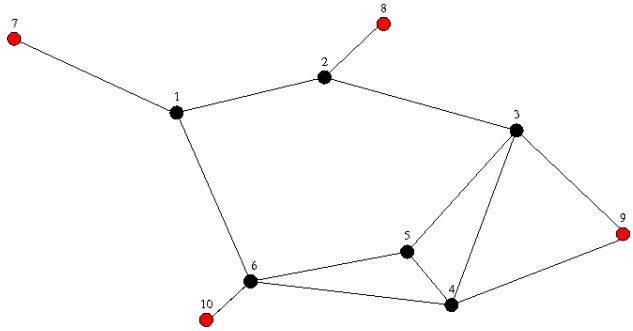


Figure 2.43: Anchor nodes (red nodes), mobile nodes (black nodes), and their communication links (edges between nodes).

We define the set $\mathcal{E}$ of communication links as

$$\mathcal{E} = \{(t,u): \text{ node } t \text{ has a communication link to node } u\}. \tag{2.71}$$

Two nodes connected by a communication link obtain a noisy measurement of their distance

$$d_{t,u} = \|\mathbf{x}_t - \mathbf{x}_u\| + \nu_{t,u}, \qquad (t,u) \in \mathcal{E}, \tag{2.72}$$

where $\nu_{t,u}$ is a zero-mean white Gaussian noise with variance $\sigma_{t,u}^2$

$$\nu_{t,u} \sim N(0, \sigma_{t,u}^2). \tag{2.73}$$

For simplicity, we assume that the distance between nodes t and u observed at node t is the same as is observed at node u:

$$d_{t,u} = d_{u,t}. \tag{2.74}$$

Conditioned on $\mathbf{x}_t$ and $\mathbf{x}_u$, $d_{t,u}$ is Gaussian with mean $\|\mathbf{x}_t - \mathbf{x}_u\|$ and variance $\sigma_{t,u}^2$

$$d_{t,u} \,|\, \mathbf{x}_t, \mathbf{x}_u \sim N(\|\mathbf{x}_t - \mathbf{x}_u\|, \sigma_{t,u}^2). \tag{2.75}$$

Given a network containing N sensors and a set $\mathbf{D}$ of observations link distances, the joint probability $p(\mathbf{X},\mathbf{D})$ can be defined as

$$p(\mathbf{X},\mathbf{D}) = p(\mathbf{x}_1,\ldots,\mathbf{x}_N,\{d_{t,u} : (t,u) \in \mathcal{E}\}) \tag{2.76}$$

$$\propto \left(\prod_{(t,u)\in\mathcal{E}} p(d_{t,u} \mid \mathbf{x}_t, \mathbf{x}_u) \right) \left(\prod_{\tau} p_\tau(\mathbf{x}_\tau) \right). \tag{2.77}$$

The goal is to develop an algorithm to approximate the joint maximum a posteriori estimate of the sensor locations $\mathbf{x}_t$ given a set of distance observations.

2.3.2 Variational Methods and Mean Field Theory

Bayesian inference provides a mathematical framework for a multitude of artificial intelligence tasks. Among these is the task of estimating position from noisy sensors. [Minka2005]

Variational methods are based on calculus of variations which covers the concept of functional derivatives. Variational methods can be applied in cases where an optimization problem requires that the quantity to be optimized is a functional. The optimal solution is found by evaluating all possible functions to find the one that maximizes, or minimizes, the functional [Bishop2006].

We seek to approximate a distribution $p(x)$ by a simpler distribution $q(x)$. To assess the approximation, a discrimination function for $p(x)$ and $q(x)$ is required. In [Minka2005], α-divergence is proposed for this task:

$$D_\alpha\big(p(x) \,||\, q(x)\big) = \frac{\int_x \alpha p(x) + (1-\alpha) q(x) - p(x)^\alpha \, q(x)^{1-\alpha} \, dx}{\alpha(1-\alpha)}. \tag{2.78}$$

Among the cases of α-divergence are:

1. Belief propagation: $\lim_{\alpha\to 1} D_\alpha\big(p(x) \,||\, q(x)\big) = KL\big(p(x) \,||\, q(x)\big)$,
2. Mean field: $\lim_{\alpha\to 0} D_\alpha\big(p(x) \,||\, q(x)\big) = KL\big(q(x) \,||\, p(x)\big)$,

where $KL(p(x)||q(x))$ is the Kullback-Leibler divergence

$$KL\big(p(x) \,||\, q(x)\big) = \int_x p(x) \ln \frac{p(x)}{q(x)} dx + \int_x \big(q(x) - p(x)\big) dx \,. \tag{2.79}$$

Here, it should be noted that the Kullback-Leibler divergence in (2.79) contains a correction factor such that it applies to unnormalized distributions. Furthermore, we stress the fact that Kullback-Leibler divergence is asymmetric with respect to $p(x)$ and $q(x)$. In this contribution, we will focus on the mean field case of α-divergence, i.e. $KL(q(x)||p(x))$.

Assume a fully Bayesian model in which all parameters are given prior distributions. The model contains both latent and observed variables. We denote the set of all latent variables $\mathbf{X}$, and the set of all observed variables $\mathbf{D}$. This probabilistic model specifies the joint probability distribution $p(\mathbf{X}, \mathbf{D})$. The objective is to approximate the posterior probability distribution $p(\mathbf{X}|\mathbf{D})$ and the model evidence $p(\mathbf{D})$. The log marginal can be decomposed as

$$\ln p(\mathbf{D}) = L(q(\mathbf{X})) + KL(q(\mathbf{X}) \,||\, p(\mathbf{X} \mid \mathbf{D})) \,, \tag{2.80}$$

where

$$L(q(\mathbf{X})) = \int_{\mathbf{X}} q(\mathbf{X}) \ln \frac{p(\mathbf{X},\mathbf{D})}{q(\mathbf{X})} d\mathbf{X} \tag{2.81}$$

is a lower bound, and

$$KL(q(\mathbf{X}) \,||\, p(\mathbf{X} \mid \mathbf{D})) = \int_{\mathbf{X}} q(\mathbf{X}) \ln \frac{q(\mathbf{X})}{p(\mathbf{X} \mid \mathbf{D})} d\mathbf{X} \tag{2.82}$$

is the Kullback-Leibler divergence for normalized distributions. The lower bound $L(q(\mathbf{X}))$ can be maximized with respect to $q(\mathbf{X})$. This equals minimizing $KL(q(\mathbf{X})||p(\mathbf{X}|\mathbf{D}))$ [Bishop2006]. $KL(q(\mathbf{X})||p(\mathbf{X}|\mathbf{D})) = 0$ when $q(\mathbf{X}) = p(\mathbf{X}|\mathbf{D})$ if any possible choice of $q(\mathbf{X})$ is allowed. Since evaluating all possible functions is an enormous (or even impossible) task, a typical approach is to restrict the range of functions over which optimization is performed. The consequence of this action is that the solution to the optimization problem becomes approximate.

Variational inference corresponds to the physics approximation framework mean field theory, when $q(\mathbf{X})$ is assumed to factorize as:

$$q(\mathbf{X}) = \prod_{i=1}^{M} q_i(\mathbf{x}_i) \tag{2.83}$$

$$= q_j(\mathbf{x}_j)\tilde{q}_j(\mathbf{x}_j), \tag{2.84}$$

where $\tilde{q}_j(\mathbf{x}_j) = \prod_{i \neq j} q_i(\mathbf{x}_i)$.

By inserting $q(\mathbf{X})$ in (2.82), it can be shown that for constrained optimization, the update equation for $q_j(\mathbf{x}_j)$ is

$$q_j^*(\mathbf{x}_j) = \underset{q_j(\mathbf{x}_j) \in Q_j}{\arg\min}\, KL(q_j(\mathbf{x}_j) \,||\, \tilde{p}_j(\mathbf{x}_j)) \tag{2.85}$$

where

$$\tilde{p}_j(\mathbf{x}_j) = \exp\left(\int_{\tilde{\mathbf{X}}} \ln p(\mathbf{X},\mathbf{D}) \left(\prod_{i \neq j} q_i(\mathbf{x}_i) \right) d\tilde{\mathbf{X}} \right) \tag{2.86}$$

$$= \exp\left(E_{\tilde{q}_j}\left[\ln p(\mathbf{X},\mathbf{D}) \right] \right). \tag{2.87}$$

Here, $\tilde{\mathbf{X}} = \mathbf{X} \setminus \{\mathbf{x}_j\}$ and $E[\cdot]$ is the expectation operator.

Now, the position of $\mathbf{x}_j$ can estimated by inserting the joint probability of latent and observed variables in (2.86) and bringing the expression to a form that matches the family Q_j.

2.3.3 Mean Field Localization Algorithm

We now derive a mean field algorithm for the localization problem defined in section 2.3.1.
To be able to derive an algorithm, we further restrict the family of approximating functions (the $q_i(\mathbf{x}_i)$'s) to be circular symmetric Gaussian distributions. Consequently, $\tilde{p}_j(\mathbf{x}_j)$ is required to belong to this family as well, and the objective is to calculate its first and second order moments. The first order moment estimates the position of $\mathbf{x}_j$, while the second order moment is the uncertainty of the position estimate.

Insertion of (2.77) into the expression for $\tilde{p}_j(\mathbf{x}_j)$ in (2.86) yields

$$\ln \tilde{p}_j(\mathbf{x}_j) = \int_{\tilde{\mathbf{X}}} \tilde{q}_j(\mathbf{x}_j) \ln p(\mathbf{X},\mathbf{D})\, d\tilde{\mathbf{X}} \tag{2.88}$$

$$\propto \int_{\tilde{\mathbf{X}}} \tilde{q}_j(\mathbf{x}_j) \left(\sum_{(t,u) \in \mathcal{E}} \ln p(d_{t,u} \mid \mathbf{x}_t, \mathbf{x}_u) + \sum_t p_t(\mathbf{x}_t) \right) d\tilde{\mathbf{X}} \tag{2.89}$$

$$\propto \ln p_j(\mathbf{x}_j) + \sum_{(t,u) \in \mathcal{E}} \int_{\tilde{\mathbf{X}}} \tilde{q}_j(\mathbf{x}_j) p(d_{t,u} \mid \mathbf{x}_t, \mathbf{x}_u) d\tilde{\mathbf{X}} + \sum_{t \neq j} \int_{\tilde{\mathbf{X}}} \tilde{q}_j(\mathbf{x}_j) \ln p_t(\mathbf{x}_t) d\tilde{\mathbf{X}} \tag{2.90}$$

We define the set of communication links that the j^{th} node share with its neighbouring nodes as:

$$\mathcal{E}_j = \{(t,u) : (t = j \text{ or } u = j) \text{ and } (t,u) \in \mathcal{E}\}. \tag{2.91}$$

Using (2.91) and neglecting constant terms, (2.90) can be rewritten as

$$\ln \tilde{p}_j(\mathbf{x}_j) \propto \ln p_j(\mathbf{x}_j) + \sum_{u \in \mathcal{E}_j} \int_{\mathbf{x}_u} \hat{q}_u(\mathbf{x}_u) \ln p(d_{j,u} \mid \mathbf{x}_j, \mathbf{x}_u) d\mathbf{x}_u \tag{2.92}$$

where $\hat{q}_u(\mathbf{x}_u)$ is the auxiliary function. From (2.92) we observe that the position estimate of node j is only depending on the prior position estimate of j, the positions of the neighbours to which j has communication links, and the distance observations to these neighbours.
By inserting

$$\hat{q}_u(\mathbf{x}_u) = \frac{1}{\sqrt{2\pi\hat{\sigma}_u^2}} \exp\left(-\frac{|| \mathbf{x}_u - \hat{\mu}_u ||^2}{2\hat{\sigma}_u^2} \right) \tag{2.93}$$

and

$$p(d_{j,u} \mid \mathbf{x}_j, \mathbf{x}_u) = \frac{1}{\sqrt{2\pi\sigma_{j,u}^2}} \exp\left(-\frac{(d_{j,u} - \| \mathbf{x}_j - \hat{\mu}_u \|)^2}{2\sigma_{j,u}^2} \right) \tag{2.94}$$

in (2.92) we see that the function $\int_{\mathbf{x}_u} \hat{q}_u(\mathbf{x}_u) \ln p(d_{j,u} \mid \mathbf{x}_j, \mathbf{x}_u) d\mathbf{x}_u$ is a Rician distribution. Calculating the first and second order moments of this distribution, inserting in (2.92) and removing terms not depending on $\mathbf{x}_j$ yields

$$\tilde{p}_j(\mathbf{x}_j) \propto p_j(\mathbf{x}_j) \exp\left(-\tfrac{1}{2} \sum_{u \in \mathcal{E}_j} \frac{1}{\sigma_{j,u}^2} \left[-2d_{j,u}\hat{\sigma}_u \sqrt{\tfrac{\pi}{2}} L_{\frac{1}{2}}\left(-\frac{\|\mathbf{x}_j - \hat{\mu}_u\|^2}{2\hat{\sigma}_u^2} \right) + \| \mathbf{x}_j - \hat{\mu}_u \|^2 \right] \right) \tag{2.95}$$

where $L_{\frac{1}{2}}(\cdot)$ is the Laguerre polynomium of order $\frac{1}{2}$.

The mean and the variance parameters that describe the position of $\mathbf{x}_j$ can be calculated by:

$$\mu_j = E_{\tilde{p}_j}[\mathbf{x}_j] \tag{2.96}$$

and

$$\sigma_j^2 = E_{\tilde{p}_j}\left[(\mathbf{x}_j - \mu_j)^2 \right] \tag{2.97}$$

Thus, we propose Algorithm 2.5 as a mean field algorithm for sensor self-localization in cooperative wireless networks.

Algorithm 2.5: Mean field localization algorithm.

1. Acquire:
   a. distance observations to neighbouring nodes, **D**
   b. observation noise variances, $\sigma_{j,u}^2$
   c. position estimates and corresponding uncertainties from neighbouring nodes, $\hat{\mu}_u$ and $\hat{\sigma}_u$.
2. Do:
   a. calculate $\tilde{p}_j(\mathbf{x}_j)$ using (2.95) and the previous estimates of μ_j and σ_j^2.
   b. calculate $\mu_{j,\text{new}}$ using (2.96),
   c. calculate $\sigma_{j,new}^2$ using (2.97)

   until $\mu_{j,\text{new}}$ has converged.

It can be shown that

$$\tilde{p}_j(\mathbf{x}_j) \to p_j(\mathbf{x}_j) \exp\left(\tfrac{1}{2} \sum_{u \in \mathcal{E}_j} \frac{1}{\sigma_{j,u}^2} \left(d_{j,u} - \left\| \mathbf{x}_j - \hat{\mu}_u \right\| \right)^2 \right) \quad \text{for} \quad \left\| \mathbf{x}_j - \hat{\mu}_u \right\| \to \infty \tag{2.98}$$

Inserting (2.70) and (2.94) in (2.77) to get the MAP solution to the localization problem reveals that applying the mean field algorithm yields MAP estimates of the position of the mobile node.

2.3.4 Numerical Results

For demonstrational purposes, the performance of the mean field localization algorithm is evaluated by Monte Carlo simulations in a highly simplified one-dimensional simulation scenario.

Two anchor nodes, $\mathbf{x}_u$ and $\mathbf{x}_v$, are fixed at respectively 3 m and 7 m from an arbitrary starting point on a straight line. The position of the mobile node, $\mathbf{x}_j$, is varied between 0 m and 10 m in steps of 1 m. This scenario is depicted in Figure 2.44.

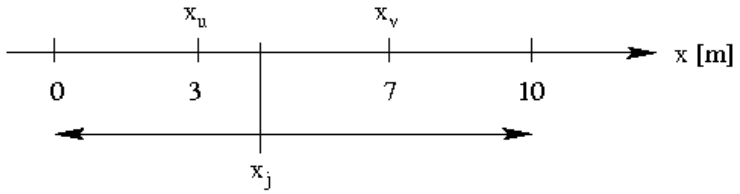


Figure 2.44: Simulation scenario: Two anchor nodes, x_u and x_v, fixed at 3 m and 7 m. The position of the mobile node, x_j, is varied from 0 m to 10 m in steps of 1 m.

In the simulations, the variance of the anchor node positions is set to 0.1 m.
The simulations are carried out as follows:
For each mobile node position, the two distances to the two anchors are calculated. To simulate observations, additive white Gaussian noise (AWGN) with variance, $\sigma_{j,u}^2$, is added to the calculated distances, and the position of the mobile node is estimated by Algorithm 2.5. To simplify matters, the noise variance is the same for both observations. The variance of the noise is changed in small steps in the range 0.25 m to 10 m for each position of the mobile node. For each mobile node position and for each realization of the noise variance, 1000 Monte Carlo experiments estimating the true position of the mobile node was made.

Figure 2.45 shows the root mean squared estimation error (RMSEE) versus the standard deviation of the noise for the mean field algorithm estimates and the MAP estimates along with the CRLB for the estimates. In Figure 2.46, the RMSEE is plotted againt the position of the mobile node.

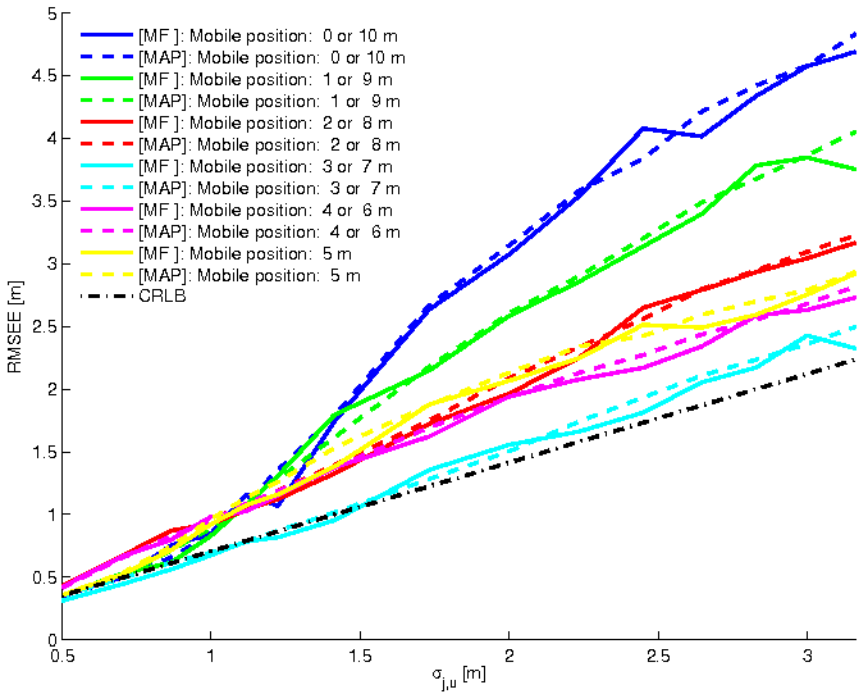


Figure 2.45: RMSEE vs. measurement noise standard deviation.

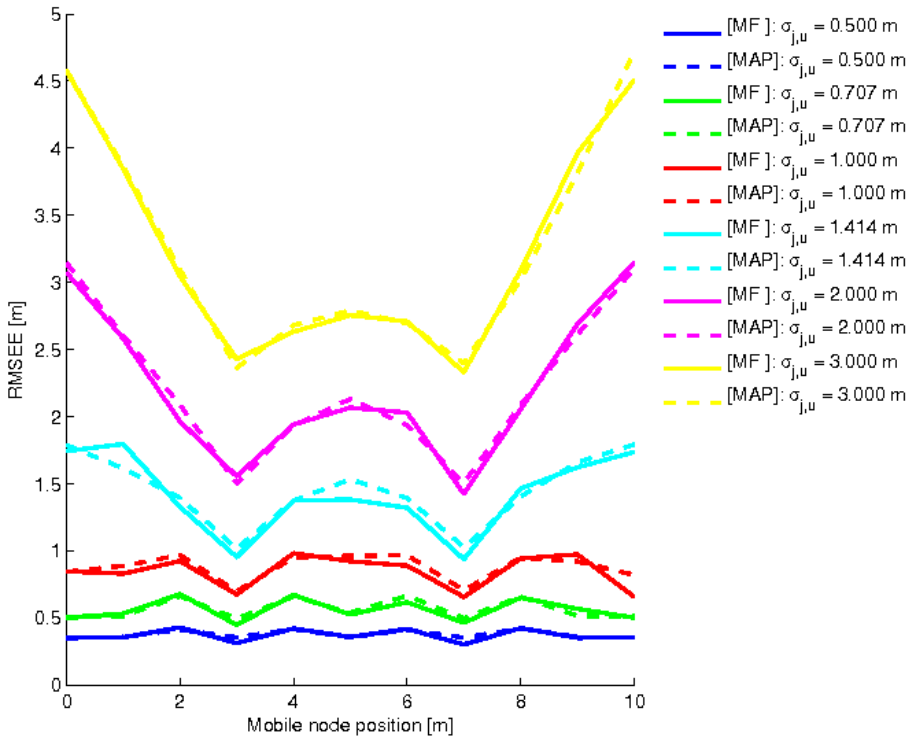


Figure 2.46: RMSEE vs. position of the mobile node.

Figure 2.45 and Figure 2.46 reveal that:

- The mean field algorithm has an overall performance that equals the MAP estimator as shown in the previous section.
- The estimation error highly depends on the position of the mobile sensor and the positions of its neighbours.
- The average estimation error is smallest when the mobile node is situated at the same position as an anchor node.

2.3.5 Conclusion and Future Work

In this contribution, we use variational methods and mean field theory to develop an algorithm for sensor self-localization in cooperative wireless networks. Using this algorithm, the position estimate of a node only depends on the node's prior knowledge on its position, the positions of the neighbour nodes and the distances observed to the neighbour nodes. We show analytically and via simulations that the mean field algorithm on average has MAP estimator performance. From the simulation results we further conclude that the estimation error highly depends on the configuration of the network nodes.

Future work includes further verifications of the localization algorithm by simulations and the use of real measurement data. Generalizing the algorithm to three dimensions and including more data (e.g. AOA) could improve the accuracy of the position estimates. Furthermore, tracking of moving mobile nodes is an interesting extension to this work.

2.4 Centralized Cooperative Positioning and Tracking

In this section, we investigate the performance of centralized cooperative positioning (CP) algorithms. Compared to traditional positioning algorithms which solely exploit ranging information from anchor nodes (ANs), CP additionally uses measurements from peer-to-peer links between the users. Since we are proposing a centralized architecture, all information has to be collected at a central entity for position calculation and further provision to the network. Hence, besides position-relevant metrics like accuracy and coverage also communications overhead and latency and their impact on the overall performance will be assessed. As we are considering a dynamic scenario, the cooperative positioning algorithms are based on extended Kalman filtering for position estimation and tracking. Simulation results for ultra-wideband (UWB) based ranging information and WiFi based communications infrastructure show the benefits of cooperative position and tracking for realistic measurement and mobility models.

In principle, there are two different procedures for CP: in the centralized approach of CP (e.g., [MRW+07][FRA07]) it is assumed that all information, i.e., the measurements collected by the mobile stations (MSs), is provided to one central entity. That could be a location server in a cellular communications system. There, the measurements are jointly processed and the position for each MS in the network is determined. Afterwards, this information can be exploited in the network or sent back to the MSs. As all measurements are processed jointly in this approach, it is the optimum procedure from a position estimation accuracy point of view. However, drawback is that all measurements have to be collected at a central entity in advance. So as to cope with scalability in dense large-scale networks or for MS-centric applications using restricted infrastructure, the distributed CP approach can also be favored as an alternative to centralized methods (e.g., [WLW09][CS09]). Here, the MSs have only the information available that they obtain from their neighbors via P2P links and the measurements with the ANs. Hence, the position estimation complexity is distributed among the MSs compared to the centralized approach. An extensive overview of CP techniques discussed under the framework of Bayesian inference can be found in [WLW09].

Generally, the communications overhead and extra-signalling is higher for cooperative approaches than for conventional (non-cooperative) positioning. Furthermore, usually the overall overhead of distributed schemes is higher than for centralized schemes. Hence, signal-processing complexity and training/signalling overhead are two key problems for existing CP approaches. This problem can be significant especially for a wireless network accommodating a large number of MSs. Therefore, an efficient CP scheme should achieve the best trade-off between communications overhead and position estimation performance.

In this section, we investigate the performance of a centralized CP scheme under realistic communications constraints and measurement models from both the positioning and the communications perspective. The centralized infrastructure is based on WiFi collecting the measurements between the ANs and the MSs as well as the P2P measurements between the MSs. The ranging is realized by UWB time-of-arrival (TOA) measurements. Additionally, mobility of the users is exploited by application of tracking algorithms based on extended Kalman filters (EKFs). Hence, simulation results will provide a realistic assessment of centralized CP in a high-mobility environment.

Throughout this section, vectors and matrices are denoted by lower and upper case bold letters, the operation $\otimes$ denotes the Kronecker product, $(\cdot)^{\mathrm{T}}$ the transpose operation, and $\mathrm{E}\{\cdot\}$ expectation. The Euclidean norm is denoted as $\|\cdot\|_2$, and the N-dimensional identity matrix is denoted as $\mathbf{I}_N$.

2.4.1 System Model

We consider N_{AN} ANs and N_{MS} MSs that are present in the scenario. The ANs are located at the known and fixed positions

$$\mathbf{x}^{(\mathrm{AN})} = \left[\mathbf{x}_1^{(\mathrm{AN}),\mathrm{T}} \quad \mathbf{x}_2^{(\mathrm{AN}),\mathrm{T}} \quad \ldots \quad \mathbf{x}_{N_{\mathrm{AN}}}^{(\mathrm{AN}),\mathrm{T}} \right]^{\mathrm{T}}, \tag{2.99}$$

where

$$\mathbf{x}_\mu^{(\mathrm{AN})} = \left[\mathbf{x}_\mu^{(\mathrm{AN})} \quad \mathbf{y}_\mu^{(\mathrm{AN})} \right]^{\mathrm{T}}, \ \mu = 1,2,\ldots,N_{\mathrm{AN}}, \tag{2.100}$$

describes the position of the AN μ. The positions of the MSs

$$\mathbf{x} = \left[\mathbf{x}_1^{\mathrm{T}} \quad \mathbf{x}_2^{\mathrm{T}} \quad \ldots \quad \mathbf{x}_{N_{\mathrm{MS}}}^{\mathrm{T}} \right]^{\mathrm{T}} \tag{2.101}$$

with

$$\mathbf{x}_\nu = \left[x_\nu \quad y_\nu \right]^{\mathrm{T}}, \ \nu = 1,2,\ldots,N_{\mathrm{MS}}, \tag{2.102}$$

have to be estimated. Note that we restrict to a two-dimensional scenario, however, an extension to three-dimensional approaches is straightforward.The range between the MS ν and the AN μ can be calculated as

$$r_{\nu,\mu}^{(\mathrm{MS-AN})}(\mathbf{x}) = \sqrt{\left(x_\mu^{(\mathrm{AN})} - x_\nu\right)^2 + \left(y_\mu^{(\mathrm{AN})} - y_\nu\right)^2} \tag{2.103}$$

and the range between the MSs ν and $\nu' \neq \nu$ is given as

$$r_{\nu,\nu'}^{(\mathrm{MS-MS})}(\mathbf{x}) = \sqrt{\left(x_\nu - x_{\nu'}\right)^2 + \left(y_\nu - y_{\nu'}\right)^2}, \tag{2.104}$$

where the dependence on the MS positions is explicitly denoted by $\mathbf{x}$. An overview of the CP principle with three ANs and two MSs is depicted in Figure 2.47.

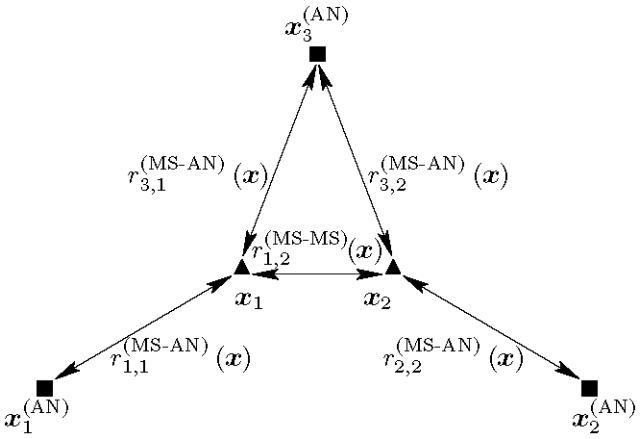


Figure 2.47: Cooperative positioning principle.

The ranging error model for the MS-AN measurements can be written as

$$\hat{r}_{\nu,\mu}^{(\mathrm{MS-AN})} = r_{\nu,\mu}^{(\mathrm{MS-AN})}(\mathbf{x}) + b_{\nu,\mu}^{(\mathrm{MS-AN})} + n_{\nu,\mu}^{(\mathrm{MS-AN})}, \tag{2.105}$$

where the bias $b_{\nu,\mu}^{(\mathrm{MS-AN})}$ and the residual noise $n_{\nu,\mu}^{(\mathrm{MS-AN})}$ depend on the LOS/NLOS status and the distance. Whereas the MS index $\nu = 1,\ldots,N_{\mathrm{MS}}$ includes all MSs in the network, the AN index for each MS $\mu = 1,\ldots,N_{\mathrm{AN,Used},\nu}$ includes only the $N_{\mathrm{AN,Used},\nu} < N_{\mathrm{AN}}$ ANs which can be used for ranging from MS ν. Equivalently, the ranging error model for the MS-MS measurements is given as

$$\hat{r}_{\nu,\nu'}^{(\mathrm{MS-MS})} = r_{\nu,\nu'}^{(\mathrm{MS-MS})}(\mathbf{x}) + b_{\nu,\nu'}^{(\mathrm{MS-MS})} + n_{\nu,\nu'}^{(\mathrm{MS-MS})}, \tag{2.106}$$

where $\nu' = 1,\ldots,N_{\mathrm{MS,Used},\nu}$ includes the available other MSs of MS ν for performing ranging. The ranging capabilities of the MSs certainly depend on channel status (LOS/NLOS, SINR, etc.) and available communications resources.

We include all available MS-AN and MS-MS measurements in the vector

$$\hat{\mathbf{r}} = \left[\hat{\mathbf{r}}^{(\mathrm{MS-AN}),\mathrm{T}} \quad \hat{\mathbf{r}}^{(\mathrm{MS-MS}),\mathrm{T}}\right]^{\mathrm{T}} \tag{2.107}$$

of dimension

$$N_{\mathrm{Used}} = N_{\mathrm{AN,Used}} + N_{\mathrm{MS,Used}} \tag{2.108}$$

with

$$N_{\mathrm{AN,Used}} = \sum_{\nu=1}^{N_{\mathrm{MS}}} N_{\mathrm{AN,Used},\nu} \tag{2.109}$$

and

$$N_{\mathrm{MS,Used}} = \sum_{\nu=1}^{N_{\mathrm{MS}}} N_{\mathrm{MS,Used},\nu}. \tag{2.110}$$

With the equivalent definitions of the range vector $\mathbf{r}(\mathbf{x})$, the bias vector $\mathbf{b}$, and the noise vector $\mathbf{n}$ with covariance matrix

$$\boldsymbol{\Sigma}_n = \begin{bmatrix} \boldsymbol{\Sigma}_n^{(\mathrm{MS-AN})} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_n^{(\mathrm{MS-MS})} \end{bmatrix}, \tag{2.111}$$

we arrive at the compact measurement model

$$\hat{\mathbf{r}} = \mathbf{r}(\mathbf{x}) + \mathbf{b} + \mathbf{n}. \tag{2.112}$$

2.4.2 Centralized Cooperative Positioning Algorithms

2.4.2.1 Static Solution

For the static solution of the centralized CP estimation problem, we follow the weighted non-linear least squares approach [KAY93][GG05] according to

$$\hat{\mathbf{x}} = \text{argmin}_{\mathbf{x}} \left(\hat{\mathbf{r}} - \mathbf{r}(\mathbf{x})\right)^{\mathrm{T}} \boldsymbol{\Sigma}_n^{-1} \left(\hat{\mathbf{r}} - \mathbf{r}(\mathbf{x})\right). \tag{2.113}$$

In the general case, there exists no closed-form solution to this non-linear $2N_{\mathrm{MS}}$ -dimensional optimization problem, and hence, iterative approaches are necessary. A standard approach to deal with this problem is based on the Gauss-Newton (GN) algorithm [KAY93]. The GN algorithm linearizes the system model about some initial value $\mathbf{x}^{(0)}$ yielding

$$\mathbf{r}(\mathbf{x}) \approx \mathbf{r}\left(\mathbf{x}^{(0)}\right) + \boldsymbol{\Phi}(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^{(0)}} \left(\mathbf{x} - \mathbf{x}^{(0)}\right), \tag{2.114}$$

with the elements of the $N_{\mathrm{Used}} \times 2N_{\mathrm{MS}}$ Jacobian matrix

$$\boldsymbol{\Phi}(\mathbf{x}) = \nabla_{\mathbf{x}}^{\mathrm{T}} \otimes \mathbf{r}(\mathbf{x}), \tag{2.115}$$

where

$$\nabla_{\mathbf{x}} = \left[\frac{\partial}{\partial x_1}, \frac{\partial}{\partial y_1}, \ldots, \frac{\partial}{\partial x_{N_{\mathrm{MS}}}}, \frac{\partial}{\partial y_{N_{\mathrm{MS}}}}\right]^{\mathrm{T}}. \tag{2.116}$$

Afterwards, the linear least squares procedure is applied resulting in the iterated solution

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \left(\boldsymbol{\Phi}^{\mathrm{T}}\left(\mathbf{x}^{(k)}\right)\boldsymbol{\Sigma}_n^{-1}\boldsymbol{\Phi}\left(\mathbf{x}^{(k)}\right)\right)^{-1} \boldsymbol{\Phi}^{\mathrm{T}}\left(\mathbf{x}^{(k)}\right)\boldsymbol{\Sigma}_n^{-1}\left(\hat{\mathbf{r}} - \mathbf{r}\left(\mathbf{x}^{(k)}\right)\right) \tag{2.117}$$

The GN algorithm provides very fast convergence and accurate estimates for good initial values. For poor initial values and bad geometric conditions the algorithm results in a rank-deficient, and thus, non-invertible matrix for certain geometric constellations of MSs and ANs.

For the considered approach, the initial value for the individual MSs is defined by the mean value of the positions of the visible ANs, i.e., corresponding to

$$\mathbf{x}_{\nu}^{(0)} = \frac{1}{N_{\mathrm{AN,Used},\nu}} \sum_{\mu=1}^{N_{\mathrm{AN,Used},\nu}} \mathbf{x}_{\mu}^{(\mathrm{AN})}. \tag{2.118}$$

2.4.2.2 Extended Kalman Filter

Usually the MSs are moving along certain tracks in the scenario. Clearly, there are strong correlations between the positions of the MSs over time. This information will be integrated in the overall position determination process and will help to improve the overall estimates in average. The Kalman filter (KF) [KAY93] is a flexible and well-known algorithm for providing such positioning estimates in the context of MS tracking applications. However, the standard KF only performs optimum if the criterions on linearity and Gaussianity are fulfilled, which is usually not the case in practical MS tracking applications. Hence, the main drawback of the linear KF is that it requires a linear state-space equation and a linear observation model (in addition to zero-mean Gaussian noise processes) to perform optimum. Clearly, for tracking only the position of the MS based on recent position estimates and the mobility model would result in such a linear relation. However, if we want to include direct range measurements that have a high non-linear property w.r.t. the current positions, the linear KF is not a reasonable approach to solve this problem.

Therefore, we propose an EKF implementation [KAY93][PP07] providing an inherent combination of CP and tracking. The EKF is based on a linearized KF and gives a good trade-off between accuracy, robustness, and complexity. The state-space and observation models are

$$\begin{aligned} \mathbf{s}[k] &= \mathbf{A}\mathbf{s}[k-1] + \mathbf{u}[k] \\ \hat{\mathbf{r}}[k] &= \mathbf{h}(\mathbf{s}[k]) + \mathbf{n}[k], \end{aligned} \tag{2.119}$$

where

$$\mathbf{s}[k] = \left[\mathbf{x}_1^{\mathrm{T}}[k] \quad \mathbf{v}_1^{\mathrm{T}}[k] \quad \mathbf{x}_2^{\mathrm{T}}[k] \quad \mathbf{v}_2^{\mathrm{T}}[k] \quad \ldots \quad \mathbf{x}_{N_{\mathrm{MS}}}^{\mathrm{T}}[k] \quad \mathbf{v}_{N_{\mathrm{MS}}}^{\mathrm{T}}[k]\right]^{\mathrm{T}} \tag{2.120}$$

is the $4N_{\mathrm{MS}}$-dimensional state-space vector in each time-step $k \in \{0,1,\ldots\}$, including two-dimensional positions and velocities of each MS as parameters that have to be estimated. The vector $\hat{\mathbf{r}}[k]$ includes the ranging measurements for each time-step and changes over time depending on the availability of the measurements. The matrix

$$\mathbf{A} = \left(\mathbf{I}_4 + \left(\begin{bmatrix} T & 0 \\ 0 & T \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \right) \right) \otimes \mathbf{I}_{N_{\mathrm{MS}}} \tag{2.121}$$

includes a priori information about the MS movements with timing updates every T time-steps. The vector $\mathbf{u}[k]$ is composed of state-space noise with diagonal covariance matrix $\mathbf{Q}$, and $\mathbf{n}[k]$ is composed of the observation noise with covariance matrix $\mathbf{\Sigma}_n[k]$. The covariance matrix can change dynamically over time depending on number and type of available measurements. Finally, the function $\mathbf{h}(\cdot)$ describes the non-linear relation between the state-space vector and the measurements.

The equations for the state-space and observation models are then used to set-up the EKF. It starts with the prediction, where knowledge of the MS movement model is applied to obtain

$$\hat{\mathbf{s}}[k \mid k-1] = \mathbf{A}\hat{\mathbf{s}}[k-1 \mid k-1], \tag{2.122}$$

with the estimate of the previous time-step $\hat{\mathbf{s}}[k-1 \mid k-1]$. Note that, e.g., the notation $[k \mid k-1]$ means that the estimate at time-step k is based on the knowledge of the measurements and the history up to time-step $k-1$. Similarly, the corresponding minimum mean square error (MMSE) matrix after that prediction step is

$$\mathbf{M}[k \mid k-1] = \mathbf{A}\mathbf{M}[k-1 \mid k-1]\mathbf{A}^{\mathrm{T}} + \mathbf{Q}. \tag{2.123}$$

Note that the EKF iterations are initialized by a static solution at the beginning. Further, we observe that in the chosen implementation the mobility of the different MSs is decoupled, i.e., for the filter equations it is assumed that the MSs move independently of each other. The Kalman gain matrix includes a weighting between the predicted estimates and the current measurements. It is given as

$$\mathbf{K}[k] = \mathbf{M}[k \mid k-1]\mathbf{H}^{\mathrm{T}}[k]\left(\mathbf{\Sigma}_n[k] + \mathbf{H}[k]\mathbf{M}[k \mid k-1]\mathbf{H}^{\mathrm{T}}[k]\right)^{-1}, \tag{2.124}$$

where – equivalent to $\mathbf{\Sigma}_n[k]$ – the dimensions can change over time. In the classical KF equations the matrix $\mathbf{H}[k]$ includes a linear relation between state-space and measurement model. Since for positioning applications we usually have a non-linear dependency, the observation equation is linearized around the predicted state-space vector, i.e.,

$$\mathbf{h}(\mathbf{s}[k]) \approx \mathbf{h}(\hat{\mathbf{s}}[k \mid k-1]) + \mathbf{H}[k]\left(\mathbf{s}[k] - \hat{\mathbf{s}}[k \mid k-1]\right), \tag{2.125}$$

where the Jacobian observation matrix is

$$\mathbf{H}[k] = \left. \frac{\partial \mathbf{h}(\mathbf{s}[k])}{\partial \mathbf{s}[k]} \right|_{\mathbf{s}[k]=\hat{\mathbf{s}}[k|k-1]}, \tag{2.126}$$

which easily can be derived from $\mathbf{\Phi}(x)$. Hence, it includes the derivations of the observation equation w.r.t. the variables of the state-space vector. Finally, the correction step combines the predicted estimates with the current measurements weighted with the Kalman gain matrix. This results in the final estimate of the state-space vector

$$\hat{\mathbf{s}}[k \mid k] = \hat{\mathbf{s}}[k \mid k-1] + \mathbf{K}[k]\left(\hat{\mathbf{r}}[k] - \mathbf{h}(\hat{\mathbf{s}}[k \mid k-1])\right). \tag{2.127}$$

The corresponding MMSE matrix is obtained as

$$\mathbf{M}[k \mid k] = \left(\mathbf{I}_{4N_{\mathrm{MS}}} - \mathbf{K}[k]\mathbf{H}[k]\right)\mathbf{M}[k \mid k-1]. \tag{2.128}$$

The EKF is designed in a flexible way, i.e., different numbers of measurements can be exploited. They also can change online for the different time-steps. Even the situation that no AN or other MS is visible for a certain time can be handled by this approach. In that situation, the movement model compensates the missing measurements.

2.4.3 Simulation Results

We start with a generic scenario, where short-range systems with fixed ANs providing ranging measurements. Additionally, we generate several simultaneous MSs that can cooperate and exchange information via the P2P links.

The positions of the 10 ANs are randomly generated in the scenario covering an area of 100m x 100m. With these ANs the MSs can perform ranging with a zero-mean Gaussian error with standard deviation of 10m. On the MS-MS links the corresponding standard deviation is 1m. Currently, we are considering 2 MSs that move through the environment for 100s. Figure 2.48 shows the scenario and the tracks of the two MSs. The visibility of the ANs for both MSs over time is depicted in Figure 2.49.

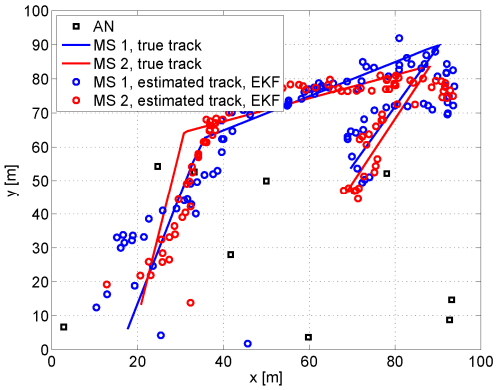


Figure 2.48: Cooperative positioning scenario, 2 MSs, 10 ANs.

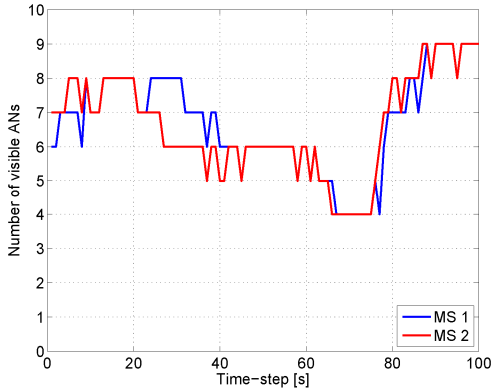


Figure 2.49: Visibility of ANs for both MSs over time.

The resulting estimation error over time for the two MSs is plotted in Figure 2.50 and Figure 2.51. We are considering both static solution and tracking with EKF as well as the conventional and cooperative approach. We observe that especially for the conventional static solution some outliers can occur. This could be a result of limited connectivity or bad geometric constellation. Clearly, an EKF tracking can reduce these errors based on the movement history and the mobility model. On the other hand, also the cooperative approach can result in an improved average position estimation error. We further observe that high gains with CP can be achieved particularly in the static case, whereas the perfomance gains for using an EKF are restricted in this scenario.

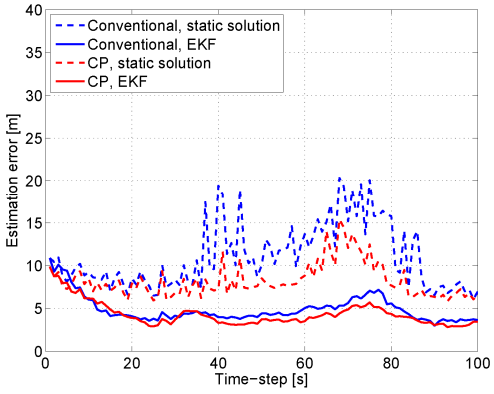


Figure 2.50: RMSE over time for MS 1.

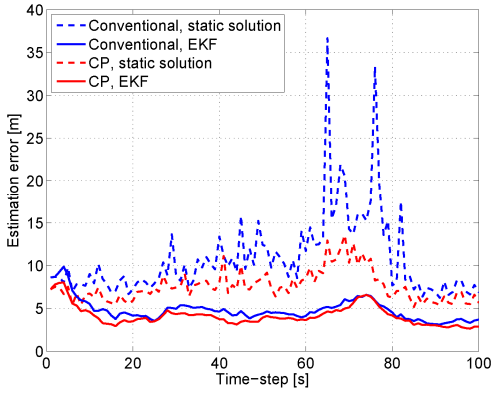


Figure 2.51: RMSE over time for MS 2.

Similar conclusions can be drawn when observing the cumulative distribution function (CDF). The performance metric CDF is defined as the probability that the absolute two-dimensional position error is below the value $\varepsilon_{\text{error}}$, i.e.,

$$\text{CDF}\left(\varepsilon_{\text{error}}\right) = \text{Prob}\left(\left\|\hat{\mathbf{x}} - \mathbf{x}\right\|_2 \leq \varepsilon_{\text{error}}\right), \tag{2.129}$$

where it was averaged over all MSs in the scenario and several noise realizations. In the considered scenario, the 90%-error is around 15m for the conventional static solution. This can be improved to 12m by allowing cooperation. When we further apply an EKF for position tracking, we can achieve a 90%-error of around 7m.

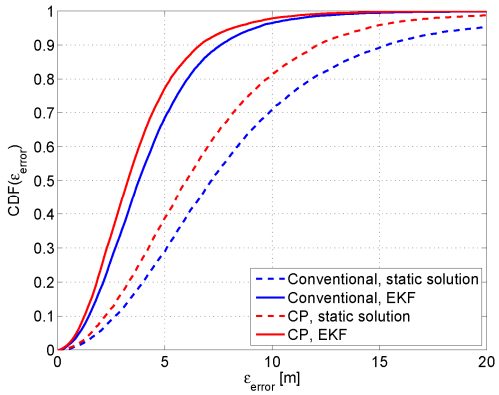


Figure 2.52: Averaged CDF for different algorithms.

Now we assume that the ANs are randomly generated with a density of 200 ANs/km² and a random coverage radius between 20m and 50m in the scenario. Furthermore, we are complementing the available measurements by wide-range cellular base stations (BSs) from a communications network with inter-site distance of 1500m that provide time difference of arrival (TDOA) measurements. The standard deviation for the TDOA measurements is set to 100m in this scenario. Here, we are considering 10 MSs that move through the environment independently.

Figure 2.53 shows the CDF for the situation, where only ANs are exploited for positioning. We observe that we can achieve a positioning accuracy smaller than 20m in 25% of the situations for the static solution (no tracking). For the EKF, we achieve 53% for positioning errors smaller than 20m. If we include the ability of CP, the performance can be increased to around 80%, i.e., the CP approach allows extending the AN coverage.

If we include additional TDOA measurements from a cellular network with BSs for the "global coverage" (cf. Figure 2.54), we achieve in 95% of the situations accuracies smaller then 20m for the conventional (non-cooperative) EKF. If we include the CP feature, this can be increased to more than 99% for this specific scenario.

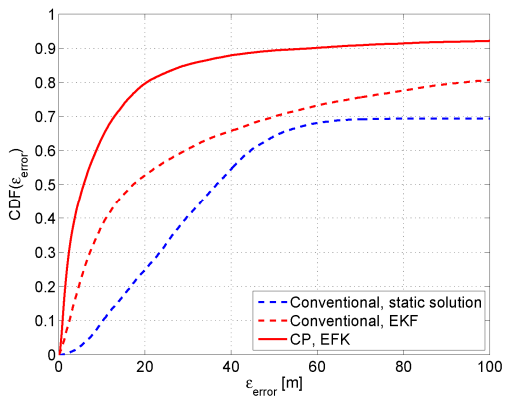


Figure 2.53: CP with ANs.

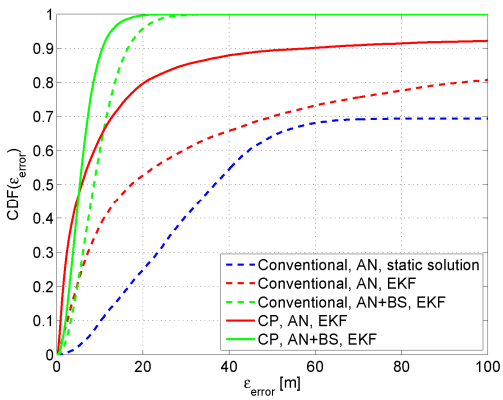


Figure 2.54: CP with ANs and BSs.

In the following, we apply realistic communications constraints to our centralized CP algorithm. As in reality, several messages have to be exchanged between the ANs and MSs as well for the P2P links, this has to be reflected in the overall performance evaluation. Hence, measurements can be outdated due to transmission delays or packets can even be lost which limits the accuracy especially in scenarios with several MSs. For details of the communications part which is based on a WiFi infrastructure for collection and providing measurements, we refer to Section 3.1. The overall 4-step evaluation methodology is depicted in Figure 2.55.

Initially, a common mobility simulation is run, which results in a trace file that describes the AN positions and MS movements according to a random waypoint group mobility model. This mobility trace is then used as a basis for simulating the message collection protocols in the ns-2 based network simulation. The output of this step is first the network-related performance metrics, and secondly this block also delivers a trace file specifying time stamps for when measurements are obtained and have been collected, according to the collection protocol. Using this trace file in combination with the mobility trace, the actual measurement values for the MS-AN and MS-MS links (including, e.g., communications

delays or packet losses) are being generated in the measurement generation block. Finally, the positioning simulation is run and positioning metrics are computed for the considered conventional and cooperative localization algorithms.

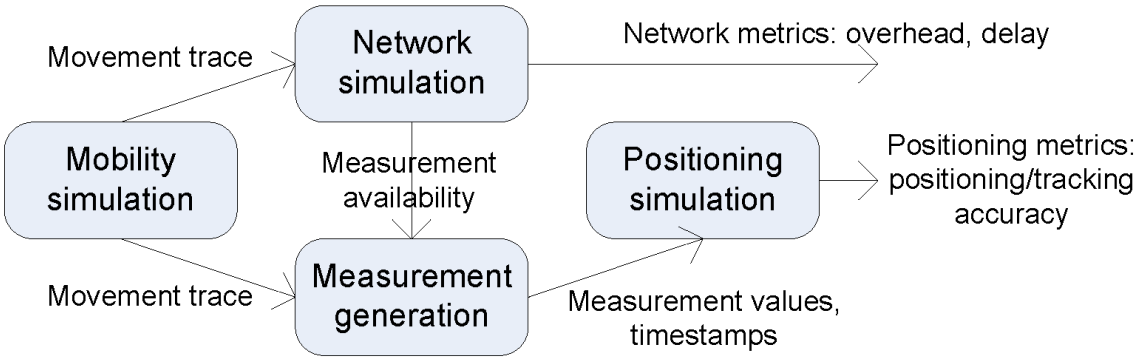


Figure 2.55: Evaluation methodology for centralized CP with realistic communications constraints.

As mobility model, we extend the random waypoint group mobility model from [D45]. In each group of MSs, one of the MSs acts as the reference node. For this MS a waypoint and speed is chosen as usual for the random waypoint model (see [D45]). For the remaining nodes in the group the same speed is used and their waypoints are chosen, so that they are randomly placed within d_{spread} of the reference MS's waypoint. An example of the resulting mobility tracks is shown in Figure 2.56. In this example there are 6 groups with 4 MSs in each group, shown with a unique color for each group.

For modeling the ranging errors, we exploit the UWB device measurements performed within the WHERE project and the derived models from that. The applied version of the model includes bias and residual noise conditioned on distance, orientation, and LOS/NLOS status of the connection. The average standard deviation of noise and the average bias are depicted in Figure 2.57 over the distance for LOS and NLOS conditions. We assume that the MS-MS connections are always LOS, whereas the MS-AN connections are NLOS in 50% of the cases.

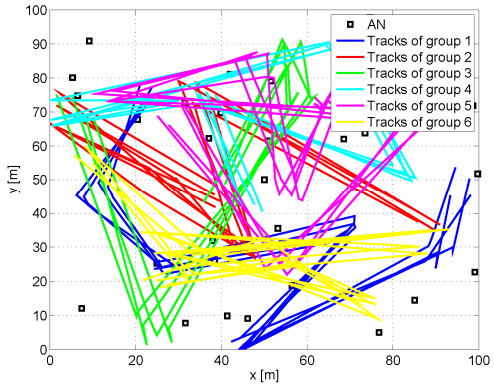


Figure 2.56: Example of group mobility simulation.

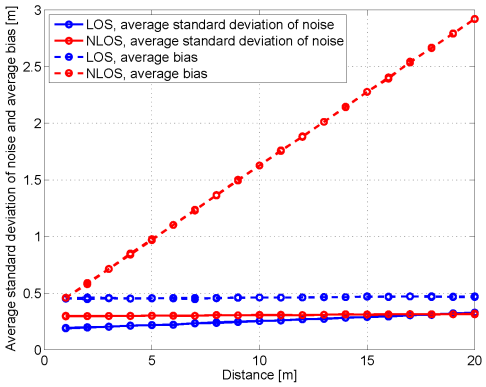


Figure 2.57: Average randing error model parameters vs. distance.

All other core simulation parameters which are more related to the positioning part are concluded in Table 2.9. More information about the communications part and the underlying parameters will be shown in Section 3.1.

Parameter	Value
Time	100 s
Size	100mx100m
Number of ANs (N_{AN})	30
Number of MS groups (N_{groups})	6
Number of MSs per group ($N_{MS/group}$)	4
Max spread relative to ref. MS in group (d_{spread})	20 m
Movement speed ($\|\mathbf{v}\|$)	2 m/s

Table 2.9: Simulation parameters.

Figure 2.58 shows the CDF for conventional (non-cooperative) and cooperative positioning for both static solution and tracking with EKF without any communications constraints. We observe that for the static solution more than 10% of the MSs cannot be localized (e.g., due to limited access to ANs or bad geometric conditions). This can be reduced by application of the EKF resulting in an error being smaller than 10m in 90% of the cases. If we allow cooperation between the MSs this can further be improved to around 3m.

Figure 2.59 includes additionally the results with the realistic communications constraints. Here, we observe that the accuracy is decreased by 1m in the conventional schemes, whereas it is reduced by around 2m and 3m for CP using static solution and EKF, respectively. As expected, the loss by communications is higher for the CP scheme compared to the conventional approach. Nevertheless, assuming CP and an EKF the 90%-error is still below 5m.

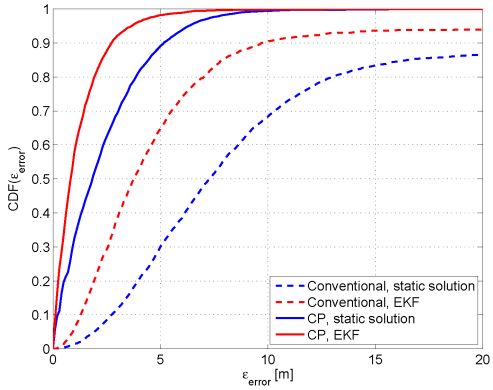


Figure 2.58: Conventional vs. cooperative positioning using static solution and EKF.

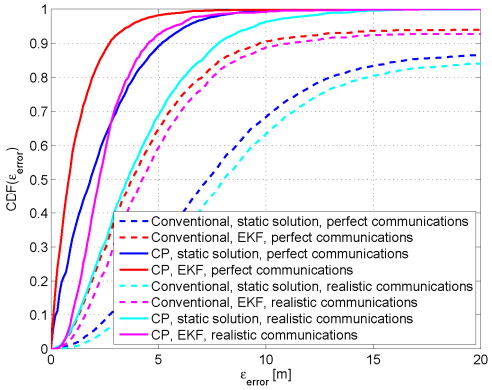


Figure 2.59: Conventional vs. cooperative positioning with realistic communications.

To evaluate the dependency on the MS-MS connectivity, in Figure 2.60 the number of MSs per group is varied. Note that an increased number of MSs per group automatically results in an increased overall number of MSs since the number of groups is kept constant. We observe that with only one MS per group no noteworthy gains can be achieved by CP compared to the conventional approach. Reason for that is that the connectivity between the groups is only limited. If we increase the number of MSs per group, e.g., to 10, cooperation can be exploited and we achieve a 90%-error of around 4m in this scenario. If we increase it further to 20, it can be seen that the performance drops down rapidly, and – in average – around 12% of the MSs cannot be localized. This could be explained by an increased communications overhead for performing CP with the 120 MSs and the resulting latency or packet-loss effects.

Figure 2.61 depicts the dependency on the MS-AN connectivity. For a low number of ANs in the scenario (e.g., 10), several MSs cannot determine their position. In that situation also the cooperation gain is restricted since overall too less ANs are available. On the other hand, if the number of ANs is too high (e.g., 70), the coverage by the ANs limits additional cooperation gains. Therefore, the number of ANs has to be chosen according to the expected MS-MS connectivity in the scenario.

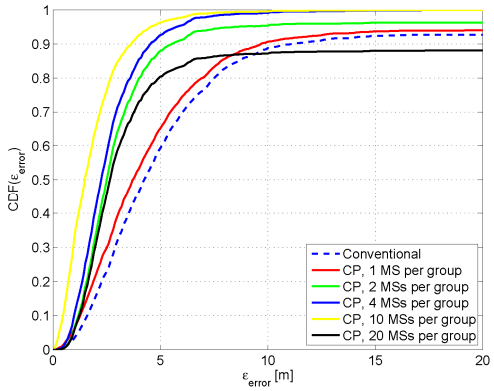


Figure 2.60: CP using EKF with realistic communications and different numbers of MS per group.

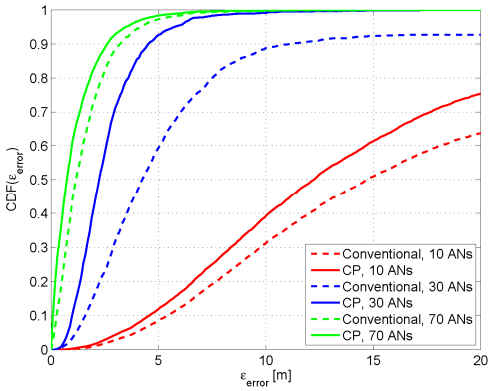


Figure 2.61: CP using EKF with realistic communications and different numbers of ANs.

2.4.4 Conclusion

In this section, we have analyzed cooperative positioning and tracking algorithms under realistic communications constraints. These constraints were modeled here based on a WiFi infrastructure and error models based on empirical measurements. It was shown that the introduction of realistic communications constraints resulted in an added delay, which had a significant effect on the positioning performance, especially for the cooperative algorithms. This is mainly due to the more complex measurement exchange that is necessary to realize the centralized cooperative positioning algorithms. We found that the static solution and the extended Kalman filter algorithms were similarly affected by the realistic communications constraints. Further, we observed that increasing the number of cooperating mobile stations had a positive impact on the positioning performance, as expected due to added cooperation possibilities. However, this was only until a tipping point was reached and the performance became worse with additional cooperating mobile stations. This tipping point is likely a result of the communication overhead becoming large, which in turn leads to increased delays. Nevertheless, in most cases the cooperative approach strongly outperforms the conventional (non-cooperative) approach.

2.5 Cooperative Localization in OFDMA-Based Cellular Networks

Localization for cellular networks is also investigated in the literature, however, most of them are non-cooperative approaches, e.g., [MSD+09]. An often utilized assumption is that a mobile terminal (MT) can receive pilot signals from at least three access points (APs), based on which each MT can estimate the location through employment of the time-difference-of-arrival (TDOA) approach. However, this assumption usually does not hold due to two practical issues: 1) Signals received from neighboring cell APs are usually very weak. This is the fact particularly for a MT located near the center of a cell. Surely, a neighboring cell AP can solve this problem by improving its transmission power. However, this would also increase the inter-cell interference. 2) The link between the home AP and an MT may be heavily shadowed so that line-of-sight (LOS) localization algorithms such as TDOA/TOA do not work properly.

For an example presented in [PIC+08], global navigation satellite system (GNSS) enabled MTs can help other MTs without GNSS to find their location. Rather than utilizing neighboring cell APs, the proposed method in [PIC +08] employs located GNSS enabled MTs to serve as anchors. The neighboring located MTs can provide enough LOS condition and signal strength for localization with acceptable performance.

This work in Section 2.5.1 and Section 2.5.2 aims to investigate cooperative localization techniques in cellular OFDMA networks. The concept of cellular network is not limited to cellphone networks, but covers a general network model which consists of several neighbouring cells, and each cell has a main AP and several fixed relay stations (FRSs). APs can have a wired link with each other. Several considered APs are connected to centralized processor, such as radio network controller (RNC), which can perform centralized data fusion. In addition, the centralized processor can be connected to received signal strength (RSS) fingerprint database as considered in WHERE WP4. The AP can receive signal from MTs to have a coarse location estimation utilizing RSS based pattern matching algorithms [SCG+05]. The considered scenario is depicted in Figure 2.62 and this scenario is suitable for both LTE and WiMax systems. As described in Figure 2.62, N_u un-located MTs are aided by N_m located MTs at the edge of an adjacent cell. The MTs communicate with their AP via their serving FRSs. The serving FRS of the un-located MTs is

referred to as the primary FRS (PFRS) and the serving FRS of the located MTs involved in the cooperation is referred to as the cooperative FRS (CFRS). By employing multiuser TOA estimation approaches, a located MT can simultaneously estimate the distances between itself and the N_u un-located MTs. Depending on the number of located MTs (i.e. N_m), our work is divided into two parts (cf. Section 2.5.1 and Section 2.5.2).

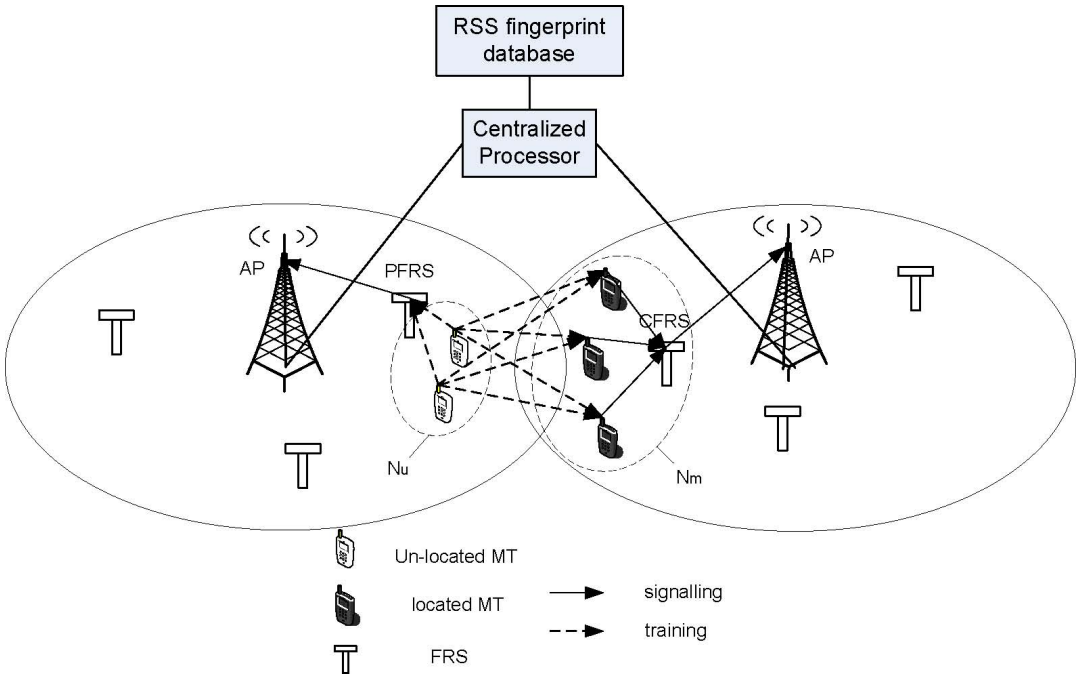


Figure 2.62：Considered scenario.

Our work about this scenario is based on the following assumptions:

Assumption 1. To enable TOA estimation for AP-MT links and MT-MT links, tight clock synchronization is assumed between APs, as well as between AP and in-cell MTs. The distance estimate is obtained through the TOA estimation.

Assumption 2. Un-located MT broadcast training signals to AP and located MT for the purpose of TOA estimation. The located MTs know the time instant of the transmitted training signals in order to successfully estimate the TOA between each un-located MT.

Assumption 3. As the un-located MTs are usually close to the PFRS and the neighboring located MT, it is reasonable to assume the PFRS and the neighboring located MT have LOS links with the un-located MT. The propagation channel for each LOS link is modeled as a single path $h = h_{LOS}$, where h_{LOS} denotes the coefficient of LOS path. All the TOA estimations are independent and the error of these estimations are modeled as zero-mean Gaussian noise, whose variance is the Cramer-Rao lower bound (CRLB) defined in [QUA81].

Assumption 4. The error of RSS-based pattern matching algorithm is independent for different un-located MTs and for different received training signal from a particular un-located MT.

Assumption 5. The error of RSS-based pattern matching algorithm is modeled as complex Gaussian noise with variance σ^2_{rss} .

Assumption 6. The locations of the located MTs are assumed perfectly known. The effect of imperfect locations of the located MTs will be investigated in future works.

For **Assumption 3,** according to [QUA81], the CRLB of the error of the TOA-based distance estimation can be expressed as

$$\sigma^2 = \frac{c^2}{8\pi^2 BT_0 (f_c^2 + B^2/12)\mathrm{SNR}} \tag{2.130}$$

where B denotes the signal bandwidth, T_0 the duration of the training signal, f_c the central carrier frequency ($f_c = 0$ for TOA estimation performed in baseband), and SNR the signal to noise ratio at the receiver. The relationship between the SNR and true link distance d is shown as follows

$$\text{SNR} = \frac{P_t}{d^{\gamma} N_o} \tag{2.131}$$

where P_t denotes the transmitted power, N_o the noise spectral density at the receiver, and γ the path loss exponent. The relationship between σ^2 and d is

$$\sigma^2 = \alpha d^{\gamma} \tag{2.132}$$

where $\alpha = \frac{c^2}{8\pi^2 B T_0 (f_c^2 + B^2/12)(P_t/N_o)}$

We denote N_{fft} and N_{cp} as the FFT size and cyclic prefix of the OFDMA system, then the system parameters are shown in Table 2.10 as follows

Parameters	Values
N_{fft}	64
N_{cp}	8
B	20MHz
T_0	3.6 μs
f_c	0
γ	2

Table 2.10: System parameters.

2.5.1 Cooperative Localization Method for Case when the Number of Located MTs is Larger than Three

In the literature [WLW09], cooperative localization in wireless networks is extensively investigated because of its capability to dramatically increase localization performance in terms of both accuracy and coverage [WLW09]. However, most of the work is done for wireless sensor networks while a few is done for cellular networks. From the communication architecture viewpoint, the MTs in sensor networks can directly communicate with each other. As a result, the distributed cooperative localization using distributed least square algorithms or message passing algorithms [WLW09] is applicable. Nevertheless, in cellular networks, MTs can not communicate directly with each other. The MTs can only communicate with the infrastructure, e.g, APs or FRSs. Thus, these cooperative localization methods can not be directly employed in the scenario investigated in [WLW09].

Cooperative localization for cellular wideband code division multiple access (WCDMA) networks was first investigated in [PIC08], where neighboring located MTs (GNSS enabled MTs) send training signals to the un-located MTs for performing RSS based range estimations. Then the location can be calculated using traditional triangulation method in [GC09]. One disadvantage of the method considered in [PIC08] is that it requires the located MTs to send their locations to the centralized processor. This location information sent by the located MTs brings significant signaling overhead to the the communication system.

The motivation for the work in Section 2.5.1 is to reduce the previously mentioned signaling overhead. Therefore, a new localization algorithm is proposed. The localization is performed firstly at CFRS to obtain the initial estimate. Then, the CFRS only needs to forward the initial estimation of the location to the AP instead of forwarding all the location information of the cooperative located MTs. By this means, less data go through CFRS-AP wireless link and the efficiency for communication is improved in terms of overhead. Therefore, the additional cost introduced by localization to communication systems is reduced.

For the following section, we consider the localization method for a single unknown user. Based on **Assumption 3,** after TOA estimation is performed, the localization method for multiuser case is a straightforward extension of the single-user case. Moreover, the localization in this method is only based on TOA estimation at located MTs and RSS fingerprint database is not required.

2.5.1.1 Proposed Method

This section presents the proposed cooperative localization method for the scenario shown in Figure 2.62 with $N_m \geq 3$. The procedure of the proposed method for each un-located MT is depicted in both Figure 2.63 and Figure 2.64. The un-located MT broadcast training signal to its neighboring located MTs and the PFRS. The PFRS and located MTs will perform TOA estimation based on the respectively received training signal. The TOA-based distance estimate between PFRS and un-located MT is obtained as $\hat{d}_{frs}$ (with true value d_{frs}) and the distance estimate between a located MT and the un-located MT is obtained as $\hat{d}_i$ (with true value d_i), $i = [1,2,...,N_m]$, where i denotes the index of cooperative link distance and N_m the number of located MTs involved in the cooperation. Then, the PFRS sends the $\hat{d}_{frs}$ to the RNC via AP, and the located MTs send their location and distance estimate $\hat{d}_i$ via assistance data (depicted in Figure 2.63) to the CFRS. Instead of further forwarding assistance data to the RNC for centralized processing, the CFRS utilizes the assistance data to perform location estimation using traditional triangulation method as in [GC09] to get the initial estimate $\hat{u}_{mt}$. $\hat{u}_{mt}$ is then sent to the RNC for final fusion together with $\hat{d}_{frs}$.

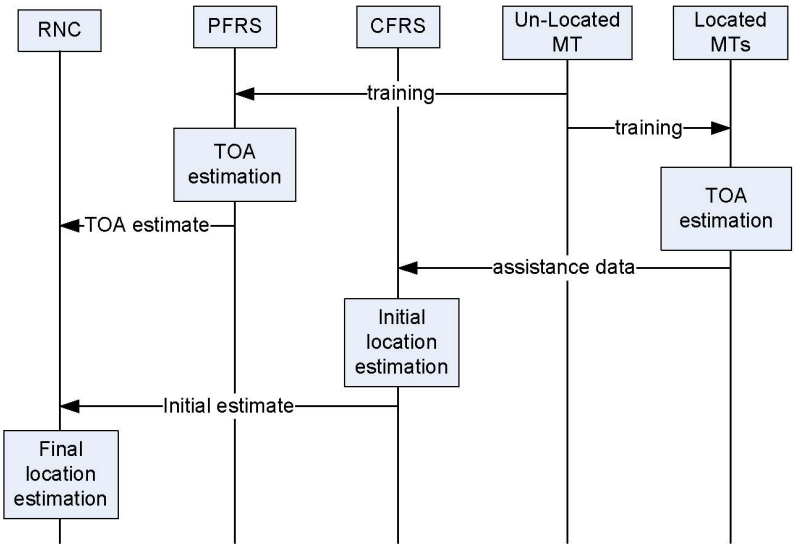


Figure 2.63: Procedure of the proposed method.

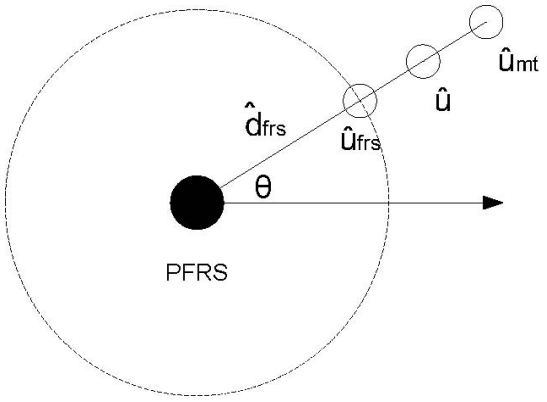


Figure 2.64: Final location estimation algortihm.

The final fusion algorithm is depicted in Figure 2.64. The circle with center point PFRS and radius $\hat{d}_{frs}$ has an intersection with the line which goes through points PFRS and $\hat{u}_{mt}$ .This intersection point is denoted as $\hat{u}_{frs}$. The final estimate $\hat{u}$ is obtained by weighted combination of the estimates $\hat{d}_{mt}$ and $\hat{d}_{frs}$ as

$$\hat{u} = f_w \hat{u}_{mt} + (1 - f_w)\hat{u}_{frs}, f_w \in [0,1] \quad (2.133)$$

where f_w is the weighting given to the initial estimate. The choice of f_w will be discussed in detail later. Note that the weighting will not change the angle of $\hat{u}$, which is obtained from merely $\hat{u}_{mt}$. The weighting only changes the amplitude of $\hat{u}$.

In order to show the advantage of the proposed method in terms of amount of transmitted data via CFRS-AP link, a simple comparison is given in Table 2.11. The traditional method refers to the case where all the location information of located MTs is sent to the centralized processor for centralized data fusion. Note that estimated location of a MT is a complex value containing a real part and a imaginary part, each of which requires 32 bits (a float). Moreover, each TOA estimate requires 32 bits. Table 2.11 shows that the transmitted data remains 64 bits for the proposed method while increasing linearly with increasing the number of located MTs for the traditional method. Figure 2.65 shows the numerical result, which shows that the proposed method significantly reduces the overhead, especially with large N_m and N_u

Conventional Method	$(64 + 32N_u)N_m$
Proposed Method	$64N_u$

Table 2.11: Signaling overhead comparison.

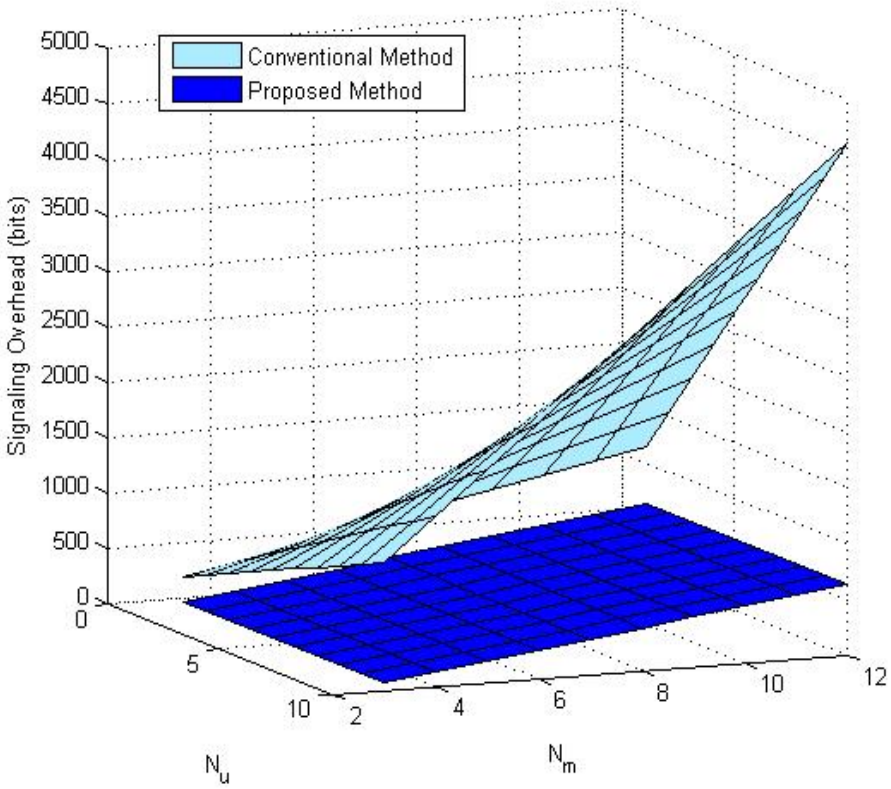


Figure 2.65：Signaling overhead between CFRS and AP.

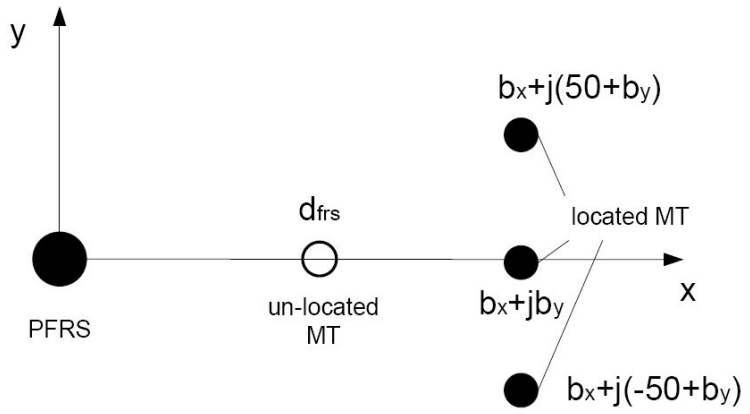


Figure 2.66: System model.

2.5.1.2 Modeling of Proposed Method

Without loss of generality, the scenario mentioned in Figure 2.62 is modeled as a system accommodating a PFRS, three located MTs (i.e. $N_m = 3$) and one un-located MT. As depicted in Figure 2.66, the x axis is the line determined by the PFRS and the unlocated MT. The PFRS is set as original point 0. The true coordinate of the un-located MT is $u = d_{frs}$. The three located MTs are placed at coordinates $b_x + j(50 + b_y)$, $b_x + jb_y$, $b_x + j(-50 + b_y)$ respectively.

Let d_i to be the true value of $\hat{d}_i$, σ_i^2 the CRLB of the error of the i th TOA-based distance estimation at located MTs, thus $\sigma_i^2 = \alpha d_i^{\gamma}$ according to (2.132). The variance of the initial location estimation is modeled using CRLB and yields

$$\mathbf{\Sigma} = \mathbf{H}^T \mathbf{C} \mathbf{H}^{-1} \tag{2.134}$$

where $\mathbf{H} = \begin{bmatrix} \frac{\partial d_1}{\partial x} & \frac{\partial d_1}{\partial y} \\ \vdots & \vdots \\ \frac{\partial d_{N_m}}{\partial x} & \frac{\partial d_{N_m}}{\partial y} \end{bmatrix}$, x and y are the real and imaginary parts of the true coordinates of un-located MT respectively, and $\mathbf{C} = \text{diag}\{\frac{1}{\sigma_1^2}, \cdots, \frac{1}{\sigma_{N_m}^2}\}$.

2.5.1.3 Performance Analysis

A simple performance analysis is given in this section to evaluate the performance of the proposed method. The metric for performance evaluation is the root mean square error (RMSE), thus the RMSE for any location estimator yields

$$\text{RMSE} = \sqrt{\frac{\sigma_x^2 + \sigma_y^2}{2}} \tag{2.135}$$

where σ_x^2 is the variance of the estimator in the direction of x axis, and σ_y^2 is the variance of the estimator in the direction of y axis. In order to model the proposed method ignoring localization algorithms for the initial estimation, we assume that the error of the initial estimation is modeled as a complex Gaussian noise, whose variance is the CRLB in [Guvenc2009]. Therefore, $\sigma_x^2 = \mathbf{\Sigma}_{11}$, $\sigma_y^2 = \mathbf{\Sigma}_{22}$. The RMSE of the initial location estimator $\hat{u}_{mt}$ yields

$$\text{RMSE}_{\hat{u}_{mt}} = \sqrt{\frac{\mathbf{\Sigma}_{11} + \mathbf{\Sigma}_{22}}{2}} \tag{2.136}$$

We will analyze the RMSE of the estimator $\hat{u}_{frs}$ first and then the RMSE of the final estimator $\hat{u}$ as follows. The analysis in this section is based on the following assumptions:

Assumption 1. The true distance between PFRS and un-located MT (i.e. d_{frs}) is large enough, and σ_y^2 of the location estimator $\hat{u}_{mt}$ (i.e. $\mathbf{\Sigma}_{22}$) is small enough. In order to describe the assumption in a mathematical way, a new metric δ is defined. According to this assumption, δ should be smaller than a threshold δ_{TH}: $\delta = \frac{\sqrt{\mathbf{\Sigma}_{22}}}{d_{frs}} < \delta_{TH}$.

Assumption 2. The involved located MTs are distributed symmetrically in the upper and lower side of x axis. Therefore, $\mathbf{\Sigma}_{12} = \mathbf{\Sigma}_{21} = 0$. The effect of unsymmetrical distribution will be evaluated via simulations. The estimator $\hat{u}_{frs}$ is determined by both $\hat{d}_{frs}$ and $\hat{u}_{mt}$.

As described in Figure 2.67, $\hat{u}_{mt}$ is distributed within the error ellipse with some probability. Based on **Assumption 2**, the error ellipse is symmetric with respect to the x axis. Furthermore, with distance estimator $\hat{d}_{frs}$, $\hat{u}_{frs}$ falls in the area between the two curves (i.e. section of two circles) with some probability. By employing the algorithm described in Figure 2.64, $\hat{u}_{frs}$ falls in the intersection of these two areas (i.e. the shadow area). The left and right edges of this shadow area are two circles, however, based on **Assumption 1**, the un-located MT is far from PFRS, i.e., the left and right edges can be approximated as two straight lines. As depicted in Figure 2.67, a straight line is determined by $\hat{d}_{frs}$ parallel to y axis, another straight line via $\hat{u}_{mt}$ and parallel to x axis is also determined. The intersection of the two straight lines is $\hat{u}_{frs}$. Therefore, for $\hat{u}_{frs}$, we approximate σ_x^2 as σ_{frs}^2 and σ_y^2 as $\mathbf{\Sigma}_{22}$. $\sigma_{frs}^2 = \alpha d_{frs}^{\gamma}$ denotes the variance of the TOA-based distance estimation at PFRS. Then the corresponding RMSE yields

$$\mathrm{RMSE}_{\hat{u}_{frs}} \approx \sqrt{\frac{\sigma_{frs}^2 + \mathbf{\Sigma}_{22}}{2}} \tag{2.137}$$

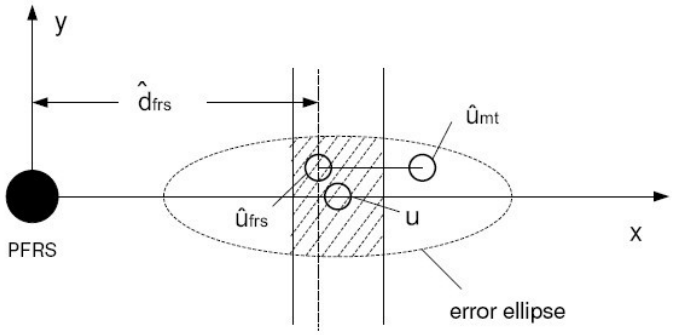


Figure 2.67: Performance analysis.

It is not difficult to observe in Figure 2.67 that the imaginary parts of $\hat{u}_{frs}$ and $\hat{u}_{mt}$ are the same, while the real part of $\hat{u}_{frs}$ and $\hat{u}_{mt}$ are independent. As a result, for $\sigma_x^2 \approx \mathbf{\Sigma}_{11} f_w^2 + \sigma_{frs}^2 (1 - f_w)^2$ and $\sigma_y^2 \approx \mathbf{\Sigma}_{22}$ the corresponding RMSE for $\hat{u}$ yields

$$\mathrm{RMSE}_{\hat{u}} \approx \sqrt{\frac{\mathbf{\Sigma}_{11} f_w^2 + \sigma_{frs}^2 (1 - f_w)^2 + \mathbf{\Sigma}_{22}}{2}}. \tag{2.138}$$

When $f_w = 0$, (2.138) reduces to (2.137). When $f_w = 1$, (2.138) reduces to (2.136). The threshold δ_{TH} and the approximation in (2.138) will be verified through simulations later. To minimize $\mathrm{RMSE}_{\hat{u}}$, the optimal weighting is chosen as

$$f_w = \frac{\sigma_{frs}^2}{\Sigma_{11} + \sigma_{frs}^2}. \tag{2.139}$$

We show in the following that with the optimal weighting, the RMSE of the proposed location estimator is equivalent to the RMSE of the conventional estimator. The RMSE can be calculated with the same principle as that of initial location estimator. To get the CRLB of the conventional estimator $\mathbf{\Sigma}'$, we only need to incorporate d_{frs} to the CRLB defined in (2.134). Based on **Assumption 2**, $\mathbf{\Sigma}'_{12} = \mathbf{\Sigma}'_{21} = 0$, where $\mathbf{\Sigma}'_{12}$ denotes the element at 1st row and 2nd column of matrix $\mathbf{\Sigma}'$, $\mathbf{\Sigma}'_{21}$ the element at 2nd row and 1st column of matrix $\mathbf{\Sigma}'$. We calculate $\mathbf{\Sigma}'_{11}$ and $\mathbf{\Sigma}'_{22}$ as follows. Since both $\mathbf{\Sigma}$ and $\mathbf{\Sigma}'$ are diagonal matrix, by incorporating d_{frs}, we have

$$\frac{1}{\mathbf{\Sigma}'_{11}} = \frac{1}{\mathbf{\Sigma}_{11}} + \frac{x^2}{\alpha d_{frs}^{\gamma+2}}. \tag{2.140}$$

As mentioned previously, x is the real part of the true coordinates of un-located MT. As defined in Figure 2.66, $x = d_{frs}$, then (2.140) can be rewritten as

$$\mathbf{\Sigma}'_{11} = \frac{\mathbf{\Sigma}_{11}\alpha d_{frs}^{\gamma}}{\mathbf{\Sigma}_{11} + \alpha d_{frs}^{\gamma}}. \tag{2.141}$$

Based on the condition $y = 0$, the same principle as for derivation of $\mathbf{\Sigma}'_{11}$ is utilized to find $\mathbf{\Sigma}'_{22} = \mathbf{\Sigma}_{22}$. For the conventional location estimator, we have $\sigma_x^2 = \mathbf{\Sigma}'_{11}$ and $\sigma_y^2 = \mathbf{\Sigma}'_{22}$. If (2.139) is plugged into σ_x^2 of proposed estimator $\hat{u}$, we can make a conclusion that σ_x^2 and σ_y^2 are the same for the conventional location estimator and the proposed location estimator respectively. The RMSE for the conventional location estimator and the proposed location estimator are exactly the same. Therefore, we can make a conclusion that with carefully choosing the located MTs in a symmetric way, the performance of the proposed method can achieve the performance of the conventional method.

2.5.1.4 Simulation Results and Discussion

For the following simulations, the parameter setting was shown in Table 2.10, which is recalled as follows. In addition, we have $P_t / N_o = 30\text{dB}$, $b_x = 300(m)$.

Parameters	Values
N_{fft}	64
N_{cp}	8
B	20MHz
T_0	3.6 μs
f_c	0
γ	2

In Figure 2.68, we investigate proposed method based on **Assumption 2.** The simulation results of the proposed method are compared with corresponding theoretical analysis. The RMSE and absolute mean bias are evaluated with different d_{frs}. Moreover, for each investigated d_{frs}, the corresponding value of δ and optimal weighting is given. It is observed from Figure 2.68 that when δ is small enough, the RMSE performance approximations match with the simulation results and the absolute mean bias is very small. The threshold δ_{TH} can be roughly set to 0.1. Figure 2.68 also shows that the value of optimal weighting increases with d_{frs} increases. The result shows that when un-located MT is closer to the located MTs, it is better to give more weight to the initial location estimate. This is because when un-located MT is close to the located MTs, the training signal power received at the located MTs increases and thus the estimation performance of $\hat{u}_{mt}$ improves.

In Figure 2.69, we investigate the proposed method based on **Assumption 1**, while we break **Assumption 2.** We fix d_{frs} at 200 (m), while b_y varies from 0 to 100 (m) to make the located MT unsymmetrically distributed. It is observed from Figure 2.69 that the simulation results still matche with the theoretical analysis. The perfect match shows that without **Assumption 2**, (2.138) are good approximations. Figure 2.68 and Figure 2.69 show that when the proposed estimators are un-biased, their RMSE matches well

with the theoretical RMSE defined in (2.138). Figure 2.69 also shows that when b_y is small, the performance of the proposed algorithm with optimal weighing can approach that of the centralized method. When $b_y = 0(m)$, the optimal performance is the same as that of the centralized method. This matches with the previous analysis.

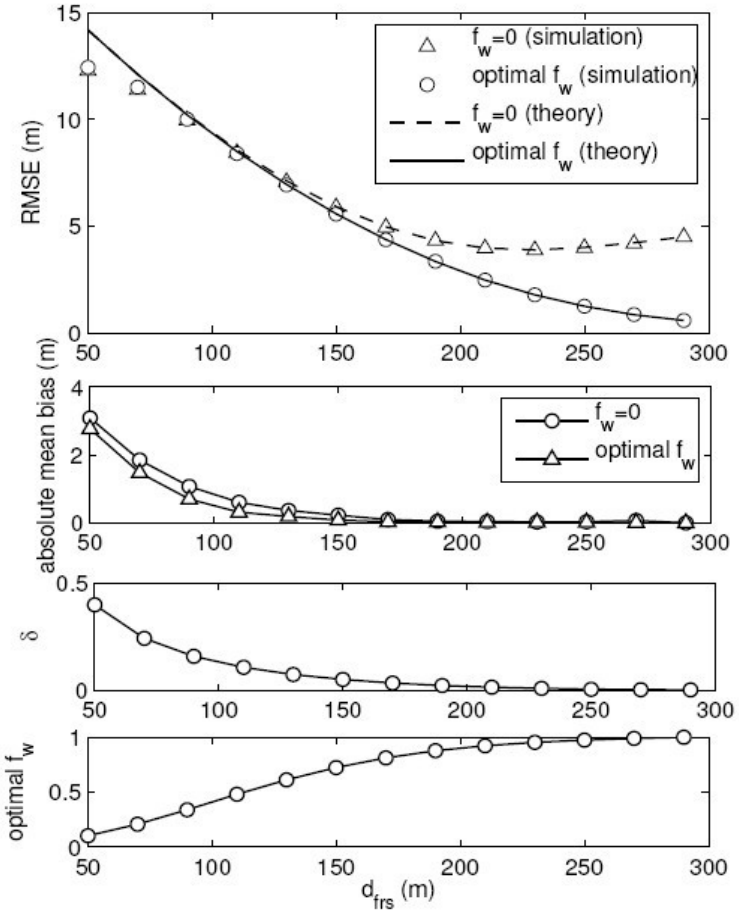


Figure 2.68: Performance approximation.

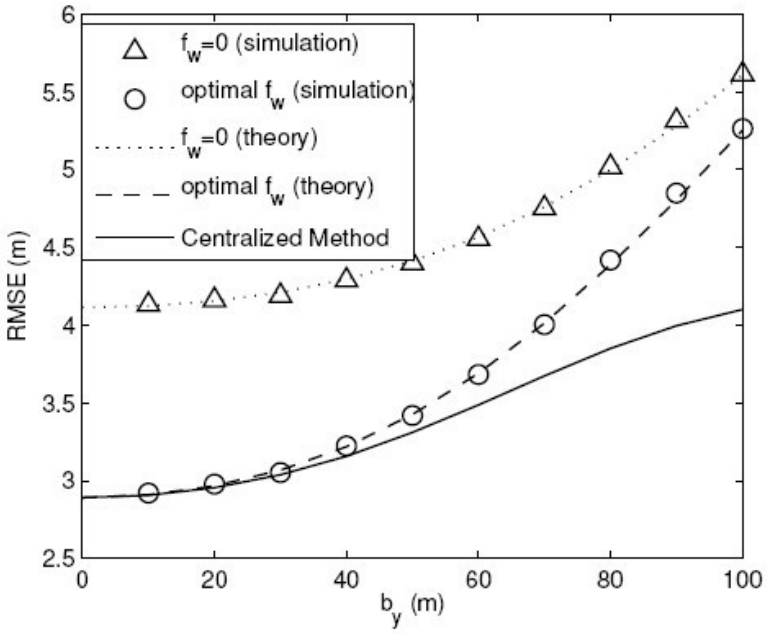


Figure 2.69: Effect of located MT distribution

2.5.2 Cooperative Localization Method for Case when the Number of Located MTs is One

The cooperative localization method proposed in [PIC08] has other limiting factors. One of these limiting factors is that the minimum number of available located MTs involved in the cooperation can not be always guaranteed. The un-located MT may not have enough neighboring located MTs to serve as anchors. Or even there are enough nearby MTs, but they are not available or not willing to help. The motivation of the work in this section is to provide new localization algorithms for the situation when the number of anchors is limited, i.e., less than three. The system model is shown in Figure 2.70, where there is only one located MT is available to help and thus the overall number of anchors is two, including PFRS. To make localization algorithms with two anchors possible, an RSS-based pattern matching method is also taken into account. In addition to TOA estimation at the PFRS and located MT, the AP can receive training signals to have a coarse location estimation utilizing RSS-based pattern matching algorithms. The coarse location estimate can remove the ambiguity of the location estimates produced by only two TOA based distance estimations.

In this section, we will propose two hybrid data fusion (HDF) based localization algorithms to fuse the estimates from RSS-based pattern matching method and TOA-based distance estimates. The investigated system model is shown in Figure 2.70, where the coordinates of the PFRS and located MT is 0 and b_x respectively. Algorithm 1 considers the case where the locations of the two un-located MTs (with true location u_1 and u_2 respectively) are seperately estimated with only one training signals. Algorithm 2 extends the Algorithm 1 to the case where the two un-located MTs cooperate. For Algorithm 2, we also consider multiple training signals to provide diversity. In order to reduce the training overhead for localization, a training ARQ scheme is proposed in Algorithm 2 by utilizing cooperative link distance estimates between two un-located MTs. The training ARQ scheme can automatically self-check a location estimate and ask for training signal resent if the estimate is not correct. Therefore, the training signal does not always need to be resent.

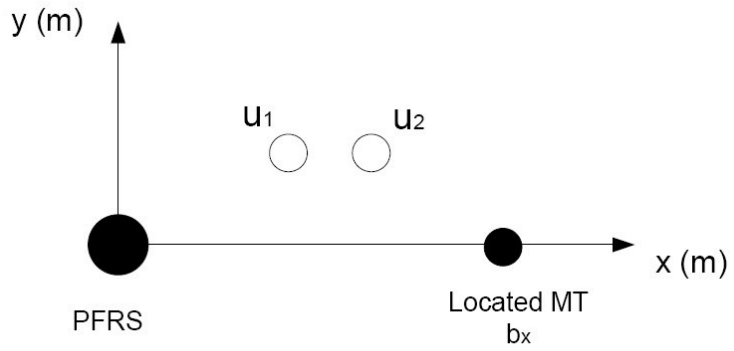


Figure 2.70: System model.

2.5.2.1 Proposed Algorithms

1) Algorithm 1

For Algorithm 1, since the locations of the two un-located MTs are separately estimated, we consider only one un-located MT, e.g., the MT with true location u_1. The un-located MT broadcasts a single training signal to the PFRS and the located MT. The PFRS receives the training signal and performs RSS-based pattern matching algorithm to obtain a coarse estimate $\hat{u}_{rss}$. In addition, both the PFRS and the located MT perform TOA based distance estimation based on this training signal. Let the distance estimate at the BS and the located MT be $\hat{d}_{frs}$ and $\hat{d}_{mt}$ respectively. As depicted in Figure 2.71, there are two circles with the true location of the PFRS and located MT as the center points, $\hat{d}_{frs}$ and $\hat{d}_{mt}$ as radius respectively. If the two circles have intersections, we denote $\hat{u}_{11}$ and $\hat{u}_{12}$ as the two intersection points. One of the intersections point is the correct location estimate, while the other one is the ambiguity. The coarse location estimate $\hat{u}_{rss}$ is used to remove the ambiguity. The basic idea is to choose the intersection point with closer distance to $\hat{u}_{rss}$ as the final location estimate. For example, if $\hat{r}_{11} < \hat{r}_{12}$, the final estimate is $\hat{u}_{11}$. The two intersections can be found by solving the following equations with respect to x and y

$$\begin{cases} x^2 + y^2 = \hat{d}_{frs}^2 \\ (x - b_x)^2 + y^2 = \hat{d}_{mt}^2 \end{cases} \tag{2.142}$$

Using the upper equation in (2.142) and subtracting the lower one to get x, then plug into the first equation in the upper one yields

$$\left(\frac{\hat{d}_{frs}^{\,2} - \hat{d}_{mt}^{\,2} + b_x^{\,2}}{2b_x} \right)^2 + y^2 = \hat{d}_{frs}^{\,2} \tag{2.143}$$

In order to have two solutions for (2.143), the following equation should be satisfied.

$$\hat{d}_{frs}^{\,2} > \left(\frac{\hat{d}_{frs}^{\,2} - \hat{d}_{mt}^{\,2} + b_x^{\,2}}{2b_x} \right)^2 \tag{2.144}$$

Then the complex locations of the two intersection point are obtained as

$$\begin{cases} \hat{u}_{11} = \dfrac{\hat{d}_{frs}^2 - \hat{d}_{mt}^2 + b_x^2}{2b_x} + j\sqrt{\hat{d}_{frs}^2 - \left(\dfrac{\hat{d}_{frs}^2 - \hat{d}_{mt}^2 + b_x^2}{2b_x} \right)^2} \\ \hat{u}_{12} = \dfrac{\hat{d}_{frs}^2 - \hat{d}_{mt}^2 + b_x^2}{2b_x} - j\sqrt{\hat{d}_{frs}^2 - \left(\dfrac{\hat{d}_{frs}^2 - \hat{d}_{mt}^2 + b_x^2}{2b_x} \right)^2} \end{cases} \tag{2.145}$$

Otherwise, if (2.144) is not satisfied, the two circles do not have intersection. In this case, we define

$\hat{u}_{11}, \hat{u}_{12}$ as

$$\hat{u}_{11} = \hat{u}_{12} = \frac{\hat{d}_{frs}^2 - \hat{d}_{mt}^2 + b_x^2}{2b_x} \tag{2.146}$$

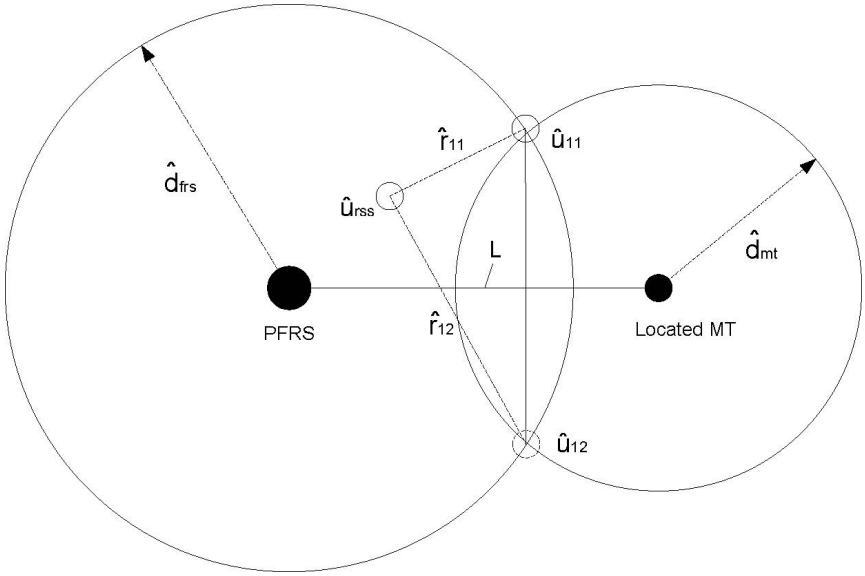


Figure 2.71: Geometrical description of Algorithm 1.

2) Algorithm 2

For Algorithm 2, we consider the case when two unlocated MTs are involved. The two MTs can send multiple training signals for the coarse location estimation using pattern matching algorithms. Theoretically, multiple training signals can provide diversity gain for both TOA estimations and pattern matching method. However, in this section, we do not consider multiple training signals for the TOA estimations. The reason is that high resolution TOA estimation algorithms are usually costly and it is not practical for the located MT to perform TOA estimation several times. Instead, we only consider the diversity gain for the coarse location estimate to remove the ambiguity. A straightforward method of utilizing the multiple training signals to remove ambiguity is:

Firstly, obtain the $\hat{u}_{rss}$ for each training signal;

Secondly, all the $\hat{u}_{rss}$ are averaged to get $\hat{u}^{(a)}{}_{rss}$;

Finally, Algorithm 1 can remove the ambiguity based on $\hat{u}^{(a)}{}_{rss}$.

Since $\hat{u}^{(a)}{}_{rss}$ is more accurate than $\hat{u}_{rss}$, the ambiguity can be removed more successfully. Theoretically, after the desire intersection is chosen by Algorithm 1, it can further be fused with the coarse location estimates to improve the final performance of location estimation. Nevertheless, this work is beyond the investigation of this section, which only focusses on ambiguity removal. The straightforward method to utilize multiple training signals can be performed separately for the two un-located MTs, thus we refer to this method as the non-cooperative approach. For the non-cooperative approach, multiple training signals are always required. This creates a lot of training overhead to the communication systems. In order to reduce the training overhead, we propose a cooperative approach in Algorithm 2 by taking into account the cooperative distance estimate between the two un-located MTs, i.e., $\hat{r}_c$ (with true value r_c), which is estimated at one of the un-located MTs based on the training set by the other un-located MT. In order to present Algorithm 2, **Step 1** and **Step 2** are firstly shown as follows.

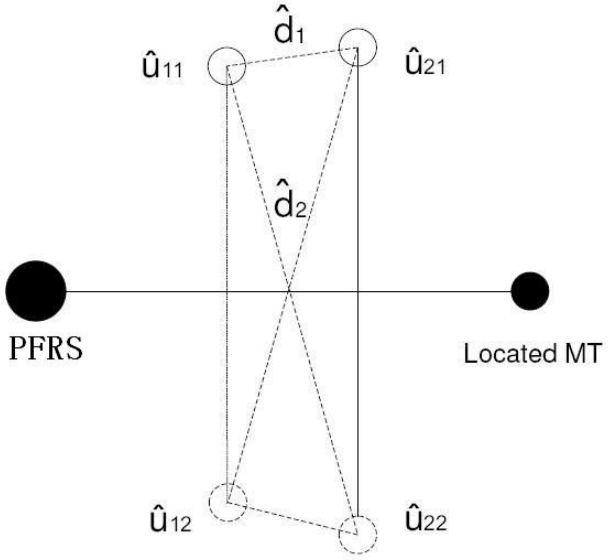


Figure 2.72：Geometrical description of Algorithm 2.

Step 1. Based on only the TOA estimations, the two intersections point are found for each un-located MTs according to (2.145) or (2.146). As shown in Figure 2.72, the four points are $\hat{u}_{11}$, $\hat{u}_{12}$, $\hat{u}_{21}$, $\hat{u}_{22}$. And $\hat{d}_1 = \|\hat{u}_{11} - \hat{u}_{21}\| = \|\hat{u}_{12} - \hat{u}_{22}\|$, $\hat{d}_2 = \|\hat{u}_{11} - \hat{u}_{22}\| = \|\hat{u}_{12} - \hat{u}_{21}\|$, where $\|\bullet\|$ denotes 2-norm operator.

Step 2. The two un-located MTs locations are obtained as a pair of location estimates $\hat{u}_1$ and $\hat{u}_2$ by employing Algorithm.1 respectively. Based on the pair of location estimates, we estimate r_c as $\hat{d}_c = \|\hat{u}_1 - \hat{u}_2\|$. A new $\hat{d}_c$ can be obtained for each training signal for the coarse location estimation.

Step 3.

1: set $T_{att} = 0$
2: **for** realization index of TOAs =1: N_t
3: perform **Step 1.** and **Step 2.**
3: **for** times=1: $T_{\max}$
4: **if** $\left\|\hat{d}_c - \hat{r}_c\right\| > d_{\mathrm{TH}}$
5: Training signal resent;
6: **repeat Step 2**;
7: $T_{att} = T_{att} + 1$;
8: **else break;**
9: **end if**
10: **end for**
11: **end for**

Table 2.12: Algorithm 2.

Then, Algorithm 2 is described in Table 2.12, where N_t is the number of realization of TOA estimations investigated in simulations. For each realization of TOA estimations (**Step 1**), **Step 2** is performed until $\hat{d}_c$ does not satisfy a criterion, i.e., $\left|\hat{d}_c - \hat{r}_c\right| > d_{\mathrm{TH}}$. d_{TH} is a threshold and defined as $d_{\mathrm{TH}} = \xi\left(\left|\hat{d}_1 - \hat{d}_2\right|\right)$, and ξ is a percentage that needs to be chosen. T_{att} describes the additional times to resend the training signal. Then, after Algorithm 2 is performed, the average time to send the training signals for coarse location estimation is $T = \frac{T_{att}}{N_t} + 1$. $T_{\max}$ is the maximum times of training signal resent. $T_{\max}$ should be small enough that training overhead will not be significantly large if the criterion is always satisfied. When all the loops in Table 2.12 are finished, the latest pair of estimates u_1 and u_2 are treated as the

final estimates of the locations.

At this stage, we explain the principle of Algorithm 2. We can realize that there are three cases for $\hat{d}_c$:
1) If the ambiguity of both of the two MTs are successfully removed, $\hat{d}_c$ is a correct estimate of r_c;
2) If the ambiguity of only one MT is removed, $\hat{d}_c$ may experience a large estimation error;
3) If the ambiguity of both of the two MTs cannot be removed, then $\hat{d}_c$ is also a correct estimate of r_c.
For example, in Figure 2.72, if we assume $\hat{u}_{11}$ and $\hat{u}_{21}$ are the correct pair of estimates, then $\hat{d}_1$ is the correct estimate of r_c. For case 1) $\hat{u}_1 = \hat{u}_{11}$, then $\hat{d}_c = \hat{d}_1$; For case 2), if the ambiguity of u_2 cannot be removed, we have $\hat{u}_1 = \hat{u}_{11}, \hat{u}_2 = \hat{u}_{22}$, then $\hat{d}_c = \hat{d}_2$, which may have a large difference with the correct estimate $\hat{d}_1$. For case 3), although $\hat{u}_1 = \hat{u}_{12}$ and $\hat{u}_2 = \hat{u}_{22}, \hat{d}_c = \hat{d}_1$, which is also correct. Since both $\hat{d}_c$ and $\hat{r}_c$ are estimates of r_c, their difference should be small enough. If case 2) happens, $\hat{d}_c$ may have larger difference with $\hat{r}_c$ compared to that of case 1) and 3). Therefore, if the criterion is satisfied, Algorithm 2 treats the pair of location estimates $\hat{u}_1$ and $\hat{u}_2$ as wrong estimates, and then automatically requires that the training signal is resent to re-estimate the coarse location of both MTs. However, Algorithm 2 may still think case 3) is correct even if the ambiguity of both of the two MTs cannot be removed.

Compared with the non-cooperative approach, the proposed cooperative approach further utilizes the cooperative distance estimate $\hat{r}_c$ to make a self-check of the coarse location estimates. As a result, training signal is not required to be resent if the estimate $\hat{d}_c$ does not satisfy the criterion. Potentially, T can be reduced compared to a fix $T = T_{\max} + 1$ for the non-cooperative scheme. The training overhead reduction is evaluated via simulation in the following section.

2.5.2.2 Simulation Results and Discussion

In this section, all the simulations follow the setting in Table 2.10 and Figure 2.70. In addition, we have $b_x = 300$ (m), $u_1 = (150, u_y)$ (m), $u_2 = (150, u_y + r_c)$ (m), $T_{\max} = 1$, $r_c = 5$ (m), $\sigma_{rss} = 20$ (m), and $N_r = 5000$. The results in Figure 2.73 show the probability P_b that (2.144) is satisfied. The P_b is evaluated with different distances between un-located MTs and the x axis, i.e., u_y. It is observed that if the un-located MT is sufficiently far away from the line linking the two anchors, the two circles have a high probability to share two different intersections. Increasing the training signal power also leads to the increase of P_b. When u_y is small, (2.144) has a high chance not to be satisfied. Then, the two intersections are the same.

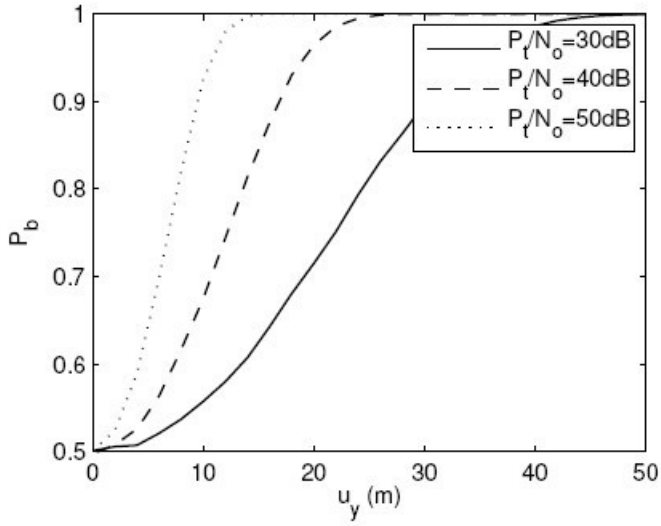


Figure 2.73: Analysis of probability of two circles to share intersections.

For the following simulations, we consider performance evaluation. The metric for the performance

evaluation is the average RMSE of the two un-located MTs. ξ should be determined to minimize the average RMSE and was set to 0.5 which was verified by simulations and utilized as a parameter for later investigations.

In Figure 2.74, we want to evaluate the effect of u_y. $P_t / N_o = 30\text{dB}$ is the additional setting for Figure 2.74. The performances of the algorithms with and without diversity are compared. It is observed from the upper plot in Figure 2.74 that with u_y increase from 0 to 100 (m), the average RMSE curves increase and then decrease. Moreover, all the algorithms share nearly the same performance when u_y is close to 0 and larger than 80 (m). The explanations are shown in three aspects as follows:
1) When u_y is close to 0: the two intersections produced by two circles are very close to each other or identical, thus imperfect ambiguity removal is likely to happen. However, the error will not be significantly large because the choice of one or the other intersection will not make too much difference or even make no difference. Further, all the algorithms share nearly the same performance;
2) When u_y increases but remains small enough (e.g. $u_y < 20(m)$): the ambiguity removal is still imperfect, but the error of the selection will lead to performance loss and the performance degrades with larger u_y ;
3) When u_y continues increasing and becomes large enough (e.g. $u_y > 20(m)$): the two intersections are far from each other, the proposed algorithms will be more likely to remove ambiguity. Therefore, all performances are improved. When $u_y > 80(m)$, both of the cooperative and non-cooperative approaches do not show diversity gain in ambiguity removal. It is because the ambiguity can be perfectly removed even with single training signals. In addition, when u_y is large enough, the performances of proposed hybrid algorithms are much better than those using only the matching pattern algorithm, which has a RMSE $= \sigma_{rss} = 20\,(\text{m})$.
We also show the training overhead in terms of T in the lower plot in Figure 2.74. It is observed that when $u_y > 30(m)$, the cooperative approach exhibits nearly the same performance as the non-cooperative approach while the overhead is the same as that of the approach without diversity. However, when u_y is too small (e.g. $u_y < 30(m)$), the cooperative approach cannot show significant benefit compared to the non-cooperative approach.

In Figure 2.75, we evaluate the effect of transmit power P_t. It is observed that with large u_y (e.g. $u_y = 100(m)$), the performance of the proposed method increases linearly when P_t / N_o increases. With small u_y (e.g. $u_y = 40(m)$), an error floor appears with P_t / N_o increases. This is because the coarse location estimate cannot perfectly remove the ambiguity with increasing transmit power. The error of RSS-based method also affects the ambiguity.

In a nutshell, the proposed algorithms can offer satisfactory performance when the un-located MTs of interest are sufficiently far away from the line linking the PFRS and the located MT. Therefore, the chosen located MT affects the performance of the proposed algorithms. Practically, the located MT can be roughly selected according to the coarse location of the un-located MT obtained from RSS or the FRS-ID of the un-located MT.

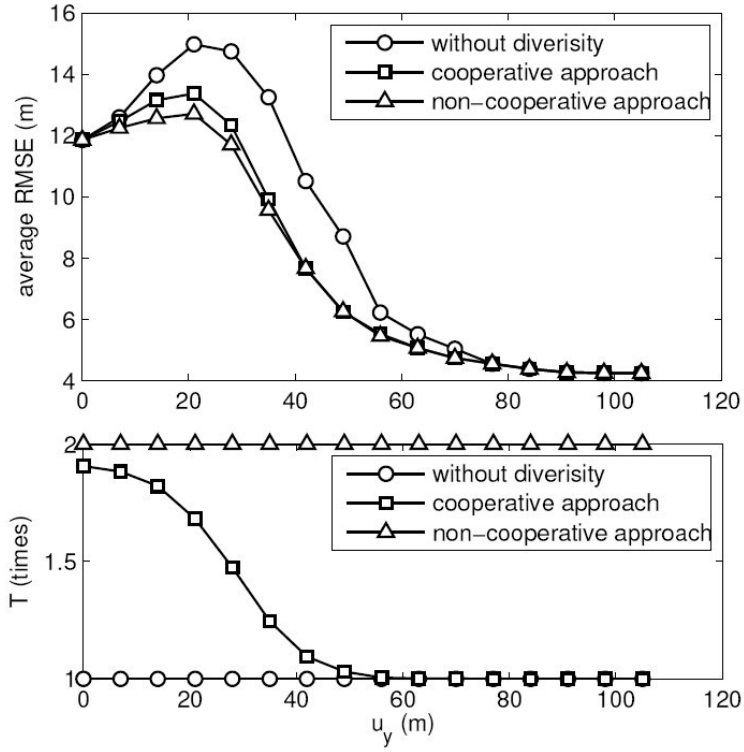


Figure 2.74: Effect of u_y .

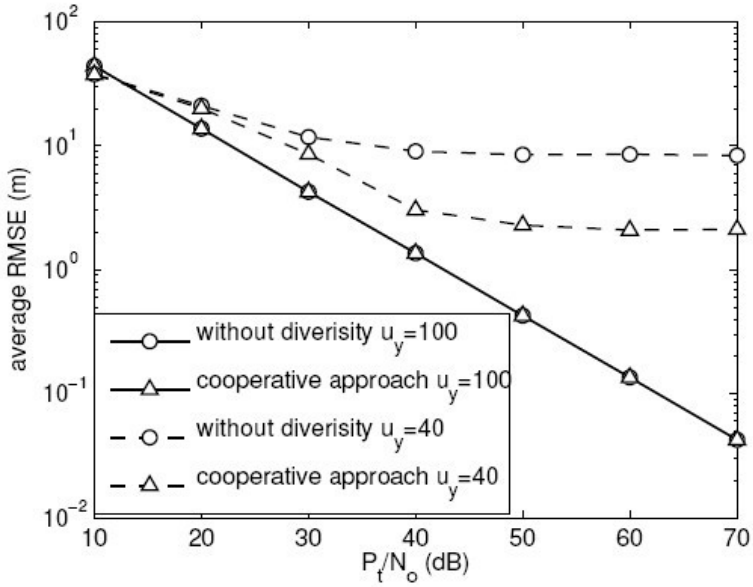


Figure 2.75: Effect of transmit power.

2.5.3 Conclusion

Section 2.5 investigates scenario T3. We proposed several cooperative localization methods and algorithms for cellular OFDMA networks. Depending on the number of available located MTs involved in the cooperation, our work is divided into Section 2.5.1 and Section 2.5.1: in Section 2.5.1, we proposed a new cooperative localization method for relay enhanced cellular OFDMA network. Compared with the centralized method performing location estimation at a centralized processor, the proposed method distributes the location estimation task at both FRS and the centralized processor. By this way, the proposed method requires less signaling overhead and thus improve the efficiency of the communication system. Moreover, the theoretical performance of the proposed method was derived. The simulation results have shown that the theoretical analysis perfectly matches the simulations when the estimators are un-biased. Furthermore, both the simulations and the theoretical analysis shown that the performance of the proposed method can achieve the performance of the centralized method, provided that the located

MTs are distributed in a roughly symmetrical manner with respect to the line linking the PFRS and the un-located MT.

In Section 2.5.2, we proposed two novel HDF cooperative localization algorithms when the number of anchors involved in the cooperation is limited to two. Algorithm 1 is suitable for independent un-located MTs, while Algorithm 2 extended Algorithm 1 to the case of two cooperative un-located MTs and with diversity from training signals. The basic idea of these two algorithms is to utilize a coarse location estimate to remove the ambiguity introduced by only two TOA-based distance estimates. By employing a training ARQ scheme, Algorithm 2 utilized the cooperative link between the two MTs to further remove the ambiguity. Algorithm 2 can operate with less training overhead compared to the non-cooperative approach utilizing diversity to remove ambiguity. The training overhead reduction was verified via simulations. Moreover, the simulation results shown that the proposed algorithms can offer satisfactory performance when the un-located MTs of interest are sufficiently far away from the line linking the two anchors.

2.6 Cooperative Localization Based on Multidimensional Scaling

2.6.1 Introduction

LMDS (Localization Based on Multidimensional Scaling) is a centralized algorithm due to Shang et al [SHA03]. Instead of using semidefinite programming, however, MDS-MAP uses a technique from mathematical psychology called multidimensional scaling (MDS). The intuition behind multidimensional scaling is simple. Suppose there are n points, suspended in a volume. We don't know the positions of the points, but we do know the distance between each pair of points. Multidimensional scaling is an $O\left(n^3\right)$ algorithm that uses the law of cosines and linear algebra to reconstruct the relative positions of the points based on the pairwise distances. Clearly, MDS has potential in the sensor localization domain. Using only ranging data, without anchors or GPS, MDS can solve for the relative coordinates of a group of sensor nodes with resilience to measurement error and rather high accuracy.

2.6.2 Localization using Multidimensional Scaling Technique

Let there be n sensors in a network, with position X_i, $i = 1 \ldots n$, and let $\mathbf{X} = \left[X_1, X_2, \ldots, X_n\right]^T$. $\mathbf{X}$ is $n \times m$ matrix, where m is the dimensionality of $\mathbf{X}$. Let $\mathbf{D} = \left[d_{ij}\right]$ be the $n \times n$ matrix of pair-wise distance measurements, where d_{ij} is the measured distance between X_i and X_j for $i \neq j$, and $d_{ii} = 0$ for all i. The distance measurements d_{ij} must obey the triangular inequality: $d_{ij} + d_{ik} \geq d_{jk}$ for all (i, j, k).The goal of MDS is to find an assignment of $\mathbf{X}$ in low-dimensional space that minimizes a cost function, defined as:

$$\mathbf{X} = \arg\min_{\mathbf{X}} F(\mathbf{X}) \tag{2.147}$$

$$F(\mathbf{X}) = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{i-1}\left(d_{ij} - \delta_{ij}\right)^2}{\sum_{i=1}^{n}\sum_{j=1}^{i-1}\delta_{ij}^2}} \tag{2.148}$$

where δ_{ij} is the real distance between X_i and X_j.

The law of cosines gives:

$$\left(X_j - X_i\right).\left(X_k - X_i\right) = \frac{1}{2}\left(d_{ij}^2 + d_{ik}^2 - d_{jk}^2\right) \tag{2.149}$$

If all measurements are perfect, then a good way to solve for the positions $\mathbf{X}$ is to choose some X_0 from $\mathbf{X}$ to be the origin of a coordinate system, and construct a matrix $\mathbf{B}$ as follows:

$$\mathbf{B} = \mathbf{X}'\mathbf{X}^T \tag{2.150}$$

with $X_i' = X_i - X_0$

We can solve for X' by taking an eigen-decomposition of $\mathbf{B}$ into an orthonormal matrix of eigenvectors ($\mathbf{V}$) and a diagonal matrix of matching eigenvalues ($\mathbf{U}$):

$$\mathbf{B} = \mathbf{X}'\mathbf{X}^T = \mathbf{U}\mathbf{V}\mathbf{U}^T \tag{2.151}$$

$$\mathbf{X}' = \mathbf{U}\mathbf{V}^{1/2} \tag{2.152}$$

We keep only the 2 (if 2D system) or 3 (if 3D system) largest diagonal values of $\mathbf{V}$ and we keep only the matching eigenvectors of $\mathbf{U}$. Then X' has the proper dimensionality. We have to apply a linear transformation of the resulting coordinates system in order to get the real one.

In real systems, there are errors. For this, MDS uses a special point in the center of the $(X_i)_i$. This point is found by "double centering" the squared distance matrix $\mathbf{D}^2$. $\mathbf{B}$ is then given by:

$$\mathbf{B} = -\frac{1}{2}\mathbf{J}\mathbf{D}^2\mathbf{J} = \mathbf{X}\mathbf{X}^T \tag{2.153}$$

$$\mathbf{J} = \mathbf{I} - \frac{1}{n}\mathbf{e}^T\mathbf{e} \tag{2.154}$$

$\mathbf{e}$ is a 1 by n vector of ones.

As before, this dimensionality reduction is done by taking an eigen-decomposition of $\mathbf{B}$, then removing eigenvalues and eigenvectors. This is a safe operation because $\mathbf{B}$ is symmetric positive definite, and therefore has n positive eigenvalues.

$$\mathbf{B} = \mathbf{X}\mathbf{X}^T = \mathbf{U}\mathbf{V}\mathbf{U}^T \tag{2.155}$$

$$\mathbf{X} = \mathbf{U}\mathbf{V}^{1/2} \tag{2.156}$$

Thus, multidimensional scaling provides a method of converting a complete matrix of distance measurements to a matching topology in 2-space or 3- space. To conclude, here are the steps of classical metric multidimensional scaling:

- Step 1 Create the symmetric matrix $\mathbf{D} = \left[d_{ij}\right]$, with $d_{ii} = 0$ and $d_{ij} + d_{ik} \geq d_{jk}$.
- Step 2 Create the symmetric matrix $\mathbf{J}$.
- Step 3 Compute $\mathbf{B}$ using $\mathbf{D}^2$ and $\mathbf{J}$.
- Step 4 Take an eigen-decomposition $\mathbf{U}\mathbf{V}\mathbf{U}^T$ of $\mathbf{B}$.
- Step 5 Let $\mathbf{V}_d$ be the diagonal matrix of the d largest eigenvalues in $\mathbf{V}$, where d is the desired dimensionality of the solution.
- Step 6 Let $\mathbf{U}_d$ be the d eigenvectors from $\mathbf{U}$ that match the eigenvalues in $\mathbf{V}_d$.
- Step 7 Compute $\mathbf{X}_d = \left[X_1, X_2, \ldots, X_n\right]^T$ using $\mathbf{X}_d = \mathbf{U}_d\mathbf{V}_d^{½}$. $\mathbf{V}_d^{½}$ can be computed by taking the square root of each of $\mathbf{V}_d$'s diagonal elements.
- Step 8 Transform the $(X_i)_i$ from $\mathbf{X}_d$ into the desired global coordinate space using some coordinate system registration algorithm. These transformed $(X_i)_i$ are the solution.

2.6.3 Simulations and Results

100 points are chosen randomly in a 50 by 50 m^2 area. In order to validate the algorithm, we suppose first that no errors are introduced on measured ranges. Then, we apply the steps described in the previous section in order to localize these points. The "no error" assumption is not realistic, for this, we assume a normal error on ranges with a standard deviation equal to 1 meter. The next two subsections plot the obtained results.

LMDS algorithm in case of no error introduced on ranges

Obviously, in the case of no error introduced on ranges the position estimates are exactly equal to the real positions. This result is shown in Figure 2.76 and it is a simple validation of the proposed algorithm.

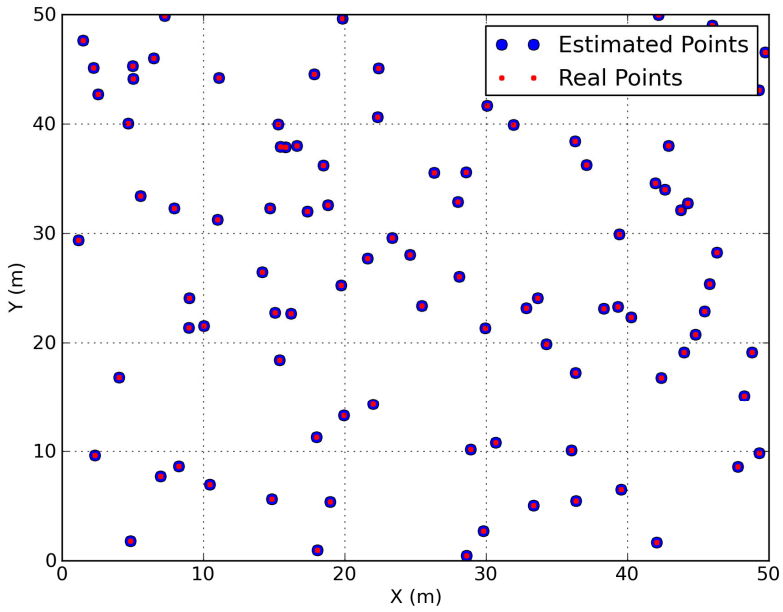


Figure 2.76: Validation of LMDS algorithm- No errors introduced on measured ranges.

LMDS algorithm in case of error introduced on ranges

In Figure 2.77, we plot the true and estimated positions using LMDS technique when an error with a standard deviation (sigma) equal to 1 meter is introduced on ranges. Even with such an imprecision on ranges, the figure shows that the LMDS technique offers a good positioning accuracy. Figure 2.78 plots the cdfs of positioning errors for different levels of ranging accuracy. **Table 2.13** summarizes the cdfs plotted in Figure 2.78. This table shows that for a ranging error with a std equal to 1 m, all points are estimated with an error less than 1.45 meter which reflects good performance. Even with a bad ranging accuracy (std = 4 m), we obtain 80 % of points which estimated with an error less than 2 meters.

Ranging error std (m)	% of positioning errors < 1 meter	% of positioning errors < 2 meter	Maximal error (m)
1.0	95	100	1.45
2.0	78	96	2.7
3.0	65	94	3.3
4.0	33	80	4.1

Table 2.13: Performances of LMDS accuracy for different ranging errors.

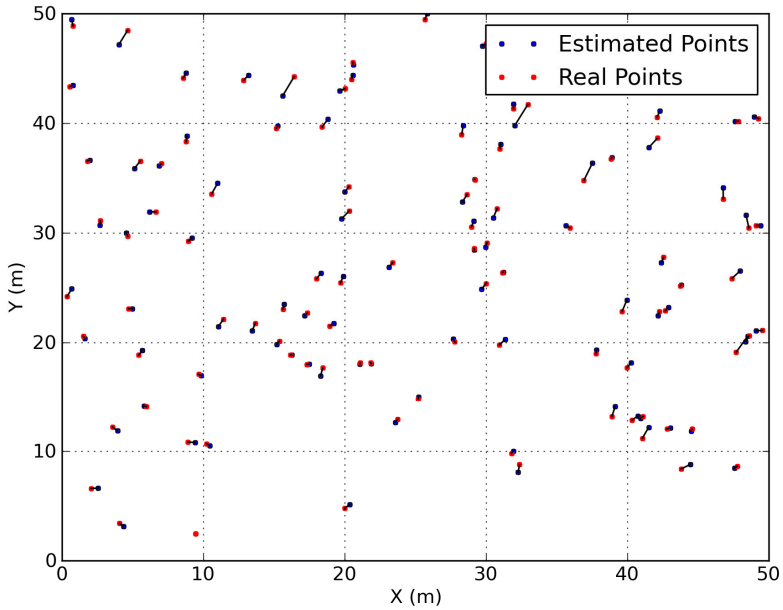


Figure 2.77: Real and estimated points using LMDS with ranging error std equal to 1 meter.

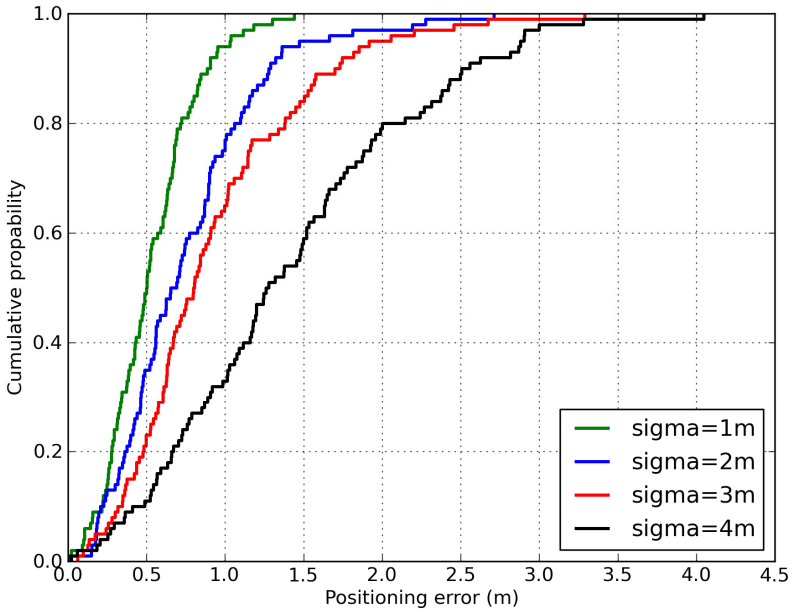


Figure 2.78: CDF of positioning error for different values of ranging error std.

Effect of ranging measurements accuracy on LMDS algorithm localization performances
In Figure 2.79, we plot the evolution of average positioning error with respect to the standard deviation of the ranging error. This figure reveals a good robustness of the LMDS technique. Indeed, for a standard deviation of ranging error equal to 10 meters, the average positioning error does not exceed 7 meters.

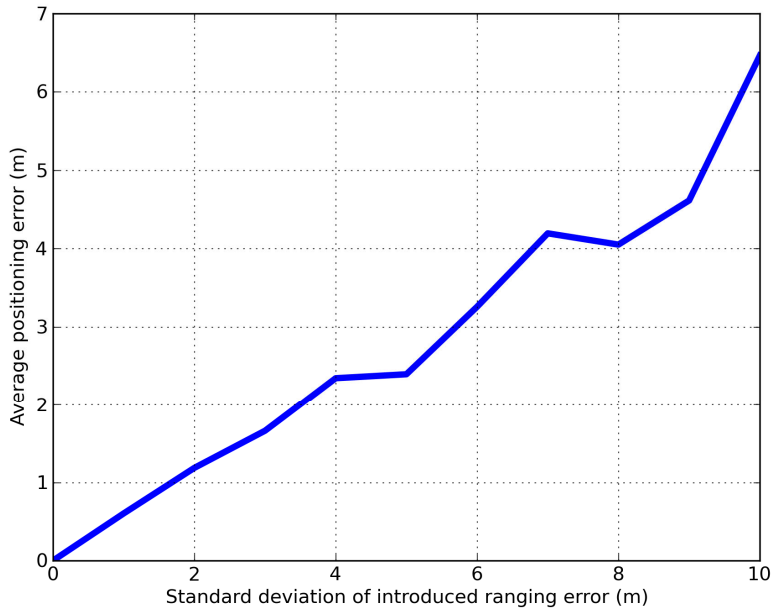


Figure 2.79: Evolution of average positioning error with respect to the introduced ranging error.

Cooperation between sensors makes overall localization accuracy better and thus enhances the positioning system. Nevertheless, it supposes the knowledge of all mutual ranges between sensors. Since TOA ranging techniques consumes additional resources, we have to reduce their uses in such cooperative algorithms. RSS is usually available with no additional cost; RSS based ranging techniques may offer good estimation of ranges between sensors once a good modelling of path loss is guaranteed. A simple scenario for cooperative localization is to assume that we have access to all RSS between all couples of sensors. Then ranges can be easily estimated using RSS ranging estimators [LAA09]. Nevertheless, some accurate TOA-based ranges may be used in order to enhance localization accuracy. This is the hybrid data fusion approach for cooperative localization.

Application of LMDS algorithm to M1 measurement campaign

In order to evaluate the proposed algorithm on M1 campaign [D41], we assume that somehow we have access to ranges information between all the pairs of transmitters. We introduce on these ranges Gaussian errors with a standard deviation equal to 1 meter. Then, we apply the LMDS technique in order to get estimated positions. In Figure 2.80, we plot the estimated and real points. The CDF of positioning error is also plotted in Figure 2.81. This last figure reveals good performances of the LMDS technique. Indeed, 85% of errors are less than 1 meter and the largest error does not reach 2.5 meters.

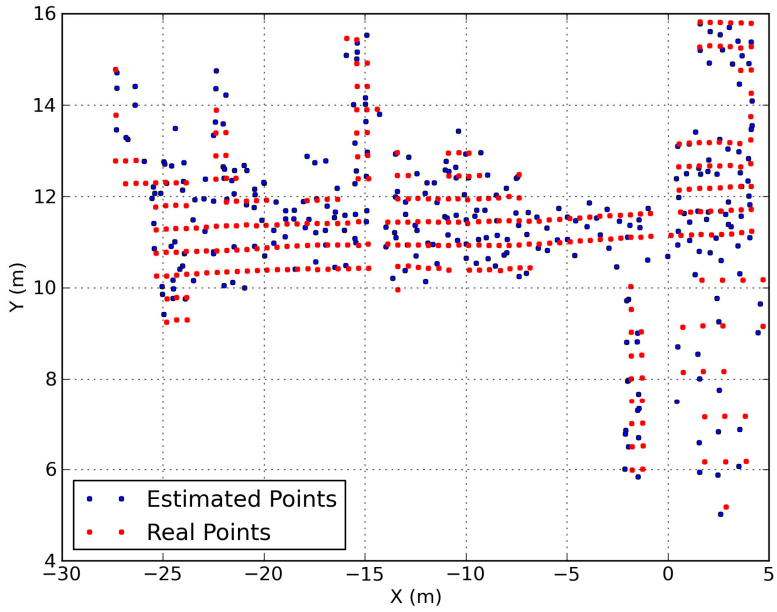


Figure 2.80: Real and estimated points in M1 environment.

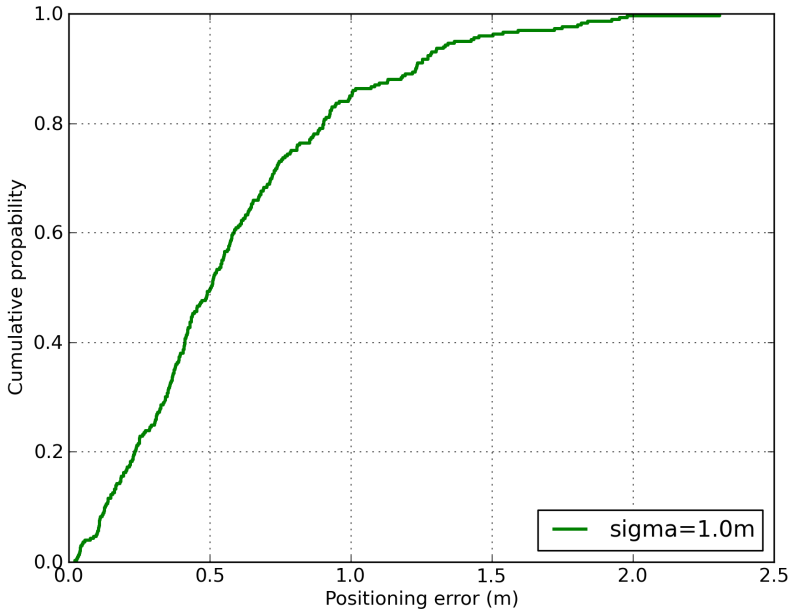


Figure 2.81: CDF of positioning error with a standard deviation of ranging error equal to 1 meter.

2.6.4 Conclusion

In this section, we have described the LMDS algorithm which is a simple algebraic algorithm for cooperative localization. The main advantages of this algorithm are simplicity, exploiting of redundant information, and ability to fuse different radio parameters. Indeed, the step of ranges collection (symmetric matrix D) can be done using different radio parameters (RSSI, TOA, or TDOA) leading to an estimation of ranges between different pairs of sensors. Nevertheless, LMDS estimates improve as ranging improves. Moreover, we may have scenarios with some missed ranges which leads to a non-complete matrix D. This may occurs when no measurement is detected between two sensors. In these cases, we may complete the matrix using some geometric relations (this solution may be heavy) or some matrix completion techniques.

2.7 Experimental Verification of Cooperation Benefits in Positioning with Biased TOA-Based Range Measurements

2.7.1 Measurement Campaign Supporting Off-Line Cooperative Positioning

In the frame of WHERE WP5 Task 5.2, a preliminary ranging measurement campaign was carried out in representative indoor environments, based on real hardware UWB Low Data Rate platforms (jointly with data transmissions at 350kbps).

Related conditional models for ranging error moments (conditioned on the actual channel configurations, actual distance and relative device orientation) were derived and reported in [D23]. These ranging models can be used for further simulations or algorithm design/evaluation purposes, accounting not only for the real channel conditions faced during the measurement campaign (e.g. NLOS obstruction or dense multipath), but also for further system contributions (e.g. residual relative clock drifts after compensation, antenna patterns, limited Rx sensibility, dynamics and time resolutions, etc…). Further details on the IR-UWB platforms and related ranging error models can be found respectively in [D53] and [D23].

For this measurement campaign, two kinds of indoor environments were investigated, namely the indoor corridor and the indoor lab. Peer-to-peer ranging transactions were performed between two devices, on a spatial grid of 1m. For each tested radio link, thousands of consecutive measurements were collected, with information about the link configuration (e.g. Initiator coordinates, responder coordinates, relative devices orientation, channel status (LOS/NLOS), physical layer parameters (e.g. modulation, synchronization, etc.)) and measurements data composed of raw 2-Way TOA-based range measurements, and corrected 3-Way TOA-based range measurements.

Although this measurement campaign was mostly intended to assess ranging performance (e.g. [D23]), it was also in part intended to support off-line cooperative positioning scenarios. Among addressed environments, only the lab shown on Figure 2.82 was considered for such cooperative positioning (e.g. See Figure 2.83), since it offers mixed LOS/NLOS channel configurations.

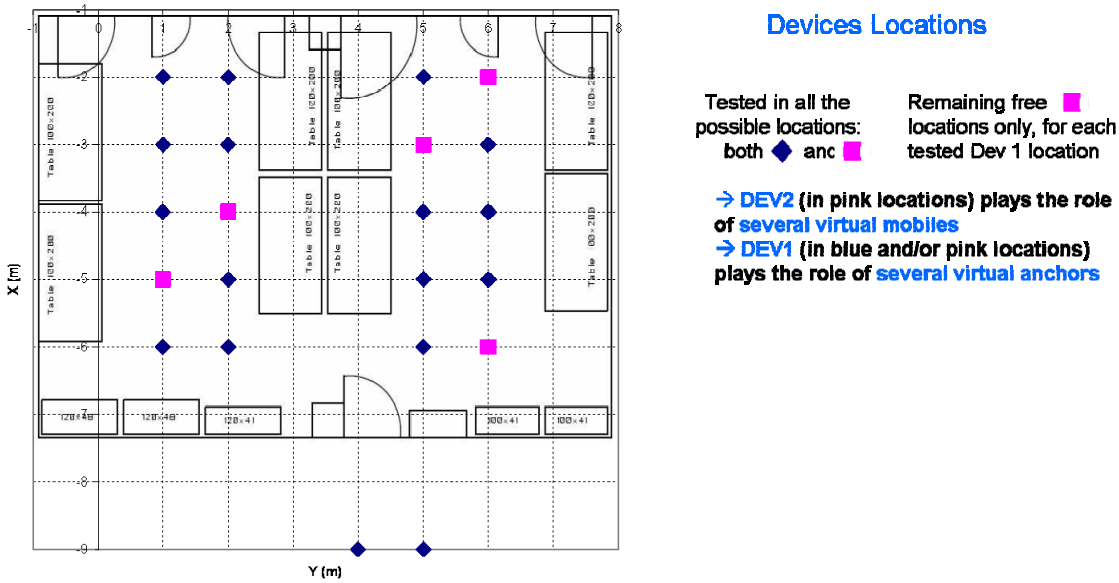


Figure 2.82: Peer-to-peer scenario (indoor Lab environment) for the emulation of cooperative positoning, with real ranging measurements issued at IR-UWB platforms in mixed LOS/NLOS configurations.

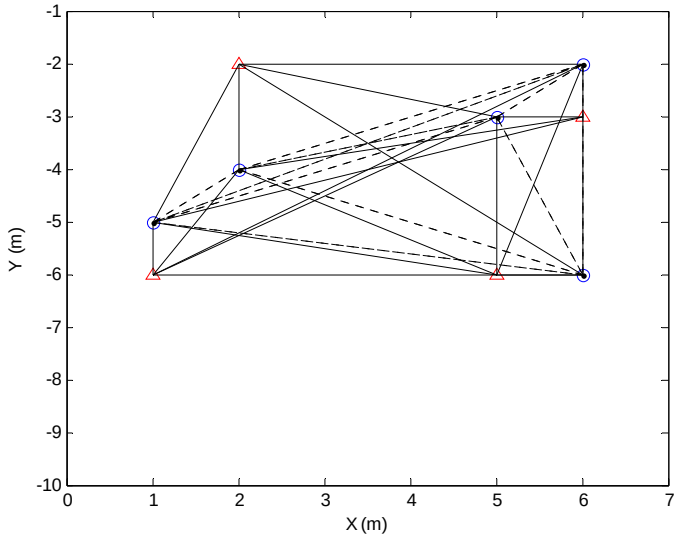


Figure 2.83: Example of off-line cooperative positioning configurations, with surrounding anchors (red triangles), mobile nodes (blue circles) and cooperative radio links (dashed lines).

2.7.2 Range Measurements and Tested Positioning Agorithms

Based on the previous measurement scenarios and data, the main purpose here is to empirically demonstrate that significant gains can already be achieved through cooperation, even with very basic or classical algorithmic approaches.

So as to perform representative location performance assessment, for a given number of mobiles and anchors, we draw randomly the locations of the cooperative mobile nodes, among all the possible locations. Then, for each of the links involved in the selected cooperative configuration, we feed the positioning algorithms with exactly the same ranging errors as that experienced during the measurement campaign, including strongly biased measurements (either due to NLOS links or harmful antenna patterns), as shown in [D23].

In the following, four main positioning algorithms are compared:

- Cooperative (Weighted) Least Squares (WLS)
- Non-cooperative (W)LS
- Cooperative (W)LS after compensating ranging biases (Genius-Aided)
- Non-cooperative (W)LS after compensating ranging biases (Genius-Aided)

For the previous solutions, standard non-linear optimization is applied with common random initial guess.

2.7.3 Off-Line Cooperative Positioning Results

On Figure 2.84 (a)–(d), we show the CDF of positioning errors for the 8 previous algorithmic options with N_a=4 fixed and surrounding anchors providing favourable Geometrical Dillution of Precision (GDOP) (See Figure 2.83), as a function of the number of cooperative mobiles N_m={2,3,4,5}, over 100 noise and bias random trials (from T5.2 measurements).

As expected, significant gains can be observed with cooperative schemes in comparison with non-cooperative approaches, whatever the tested optimization procedure (i.e. WLS and LS, biased or unbiased). The gain is even more significant:

- when the number of mobile nodes increases, hence providing more numerous cooperative measurements (i.e. information redundancy and more spatial diversity),
- with respect to nodes suffering from high location errors within non-cooperative schemes (in the larger error regime, e.g. at CDF = 95%).

Another remark is that unbiased Weighed Least Squares strategies suffer more in the larger-error regime, where nodes are subject to strongly biased range measurements, which are no more compatible with the assumed centered underlying error models.

On Figure 2.85, we show the CDFs of positioning errors for WLS approaches only, comparing different anchors configurations (i.e. fixed surrounding anchors, as previously, vs. all the possible anchors configurations under the same number of references), illustrating the impact on location performances of particular GDOP conditions at mobile nodes.

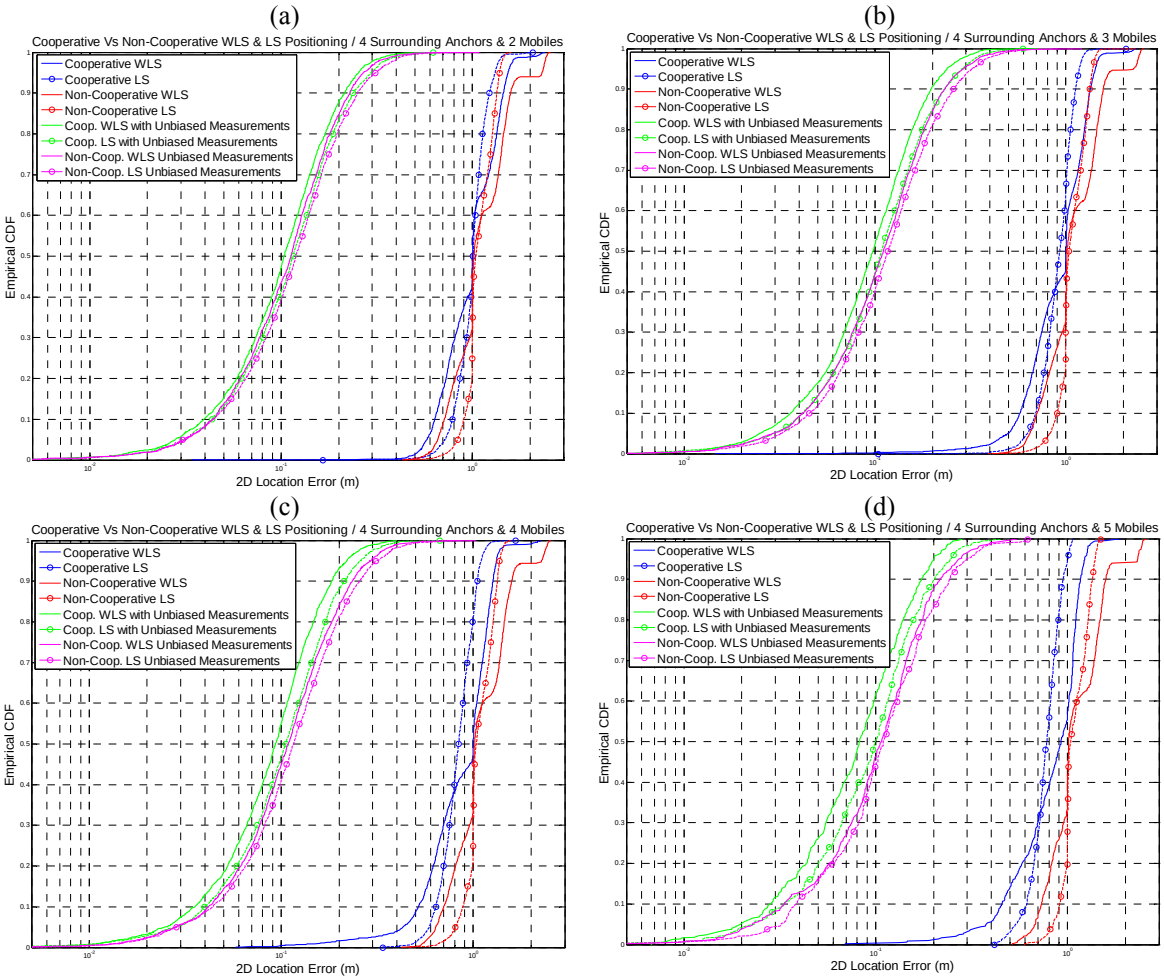


Figure 2.84: CDF of 2D positioning errors for cooperative and non-coopertaive (W)LS positioning, with 4 fixed and surrounding anchors, as a function of the number of cooperative mobiles.

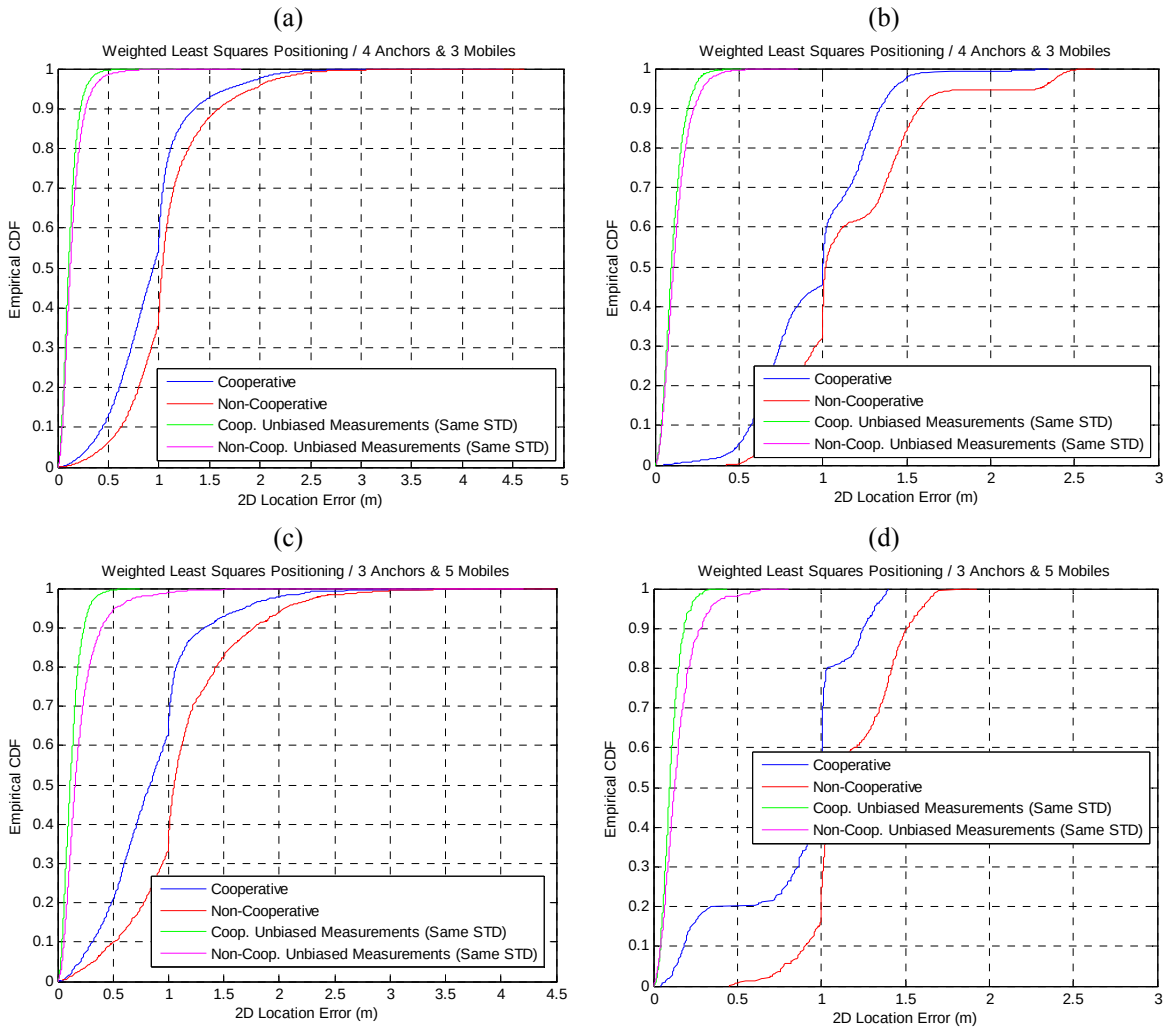


Figure 2.85: CDFs positioning error for cooperative or non-cooperative WLS positioning, with exhaustive testing over all the possible combinations of 3/4 anchors on the scene (a & c), vs. a fixed configuration with 3/4 surrounding anchors (b & d), as previously.

2.7.4 Conclusion and Perspectives

In this section, based on real data issued at WP5 hardware platforms, we have practically illustrated the benefits from mobile-to-mobile cooperation with respect to positioning precision, based on real data issued at WP5 hardware platforms. This gain is verified in harmful environments that would adversely lead to strongly biased measurements. In comparison with non-cooperative approaches, it is even more remarkable in case of generalized biased measurements (either due to NLOS links or harmful antenna patterns). We have also illustrated the importance of GDOP effects, which clearly play a role in the final location performance (through the relative geometrical configuration of both virtual anchors and other mobiles here) and hence might locally alter cooperation gains for certain mobile nodes. This remark tends to justify the general efforts made in WP2 T2.2 aiming at more advanced links selection/discarding schemes identifying the most relevant cooperative measurements and neighbours. Finally, it has been shown that the gain on the location error after removing systematic biases could be spectacular, emphasizing the weakness of simple LS approaches in case of NLOS situations and the needs for more advanced bias mitigating techniques in trackingor positioning, as proposed in most of the solutions put forward in WHERE WP2 and foreseen in WHERE2 WP2. Note that the later project will enable to validate even further the actual gain from cooperation in more complex environments, under mobility and network heterogeneity.

3. Communication Aspects of Cooperative Positioning

3.1 Realistic Communications Constraints for Centralized Cooperative Positioning and Tracking

In this section we describe and evaluate network related performance metrics of the realistic communication constraints for the conventional and cooperative localization algorithms described in section 2.4. Specifically, we consider the timing of measurement exchanges and availability of measurements through accurate simulations, and a group mobility model that mimics correlated user movements.

For both the conventional and the cooperative approaches for localization that are considered in this work we have defined protocols that are responsible for collection of measurements and provision of a location estimate. In the following we describe these protocols. In addition to collecting the measurements in the localization server, we assume that the location estimate is needed by an application at the MS, which polls the location every μ_{loc} seconds.

3.1.1 Measurement Collection for Device-Based Conventional Localization

In this case, the localization algorithm uses only measurements from the MS-AN links as sketched in Figure 3.1 which shows an example scenario with 4 ANs. As the MS holds all measurements necessary to compute the location estimate, we assume the localization/tracking algorithm is run in the MS.

Link measurements are obtained from IEEE 802.11 MAC beacons that are being broadcast in an unsynchronized manner from the ANs every μ_{beacon} seconds. Further, we assume that the transmit power P_{tx} is fixed, known and equal for all ANs. Depending on the transmit power level and the density of ANs used in a given scenario, the number of ANs within communication range of the MS and hence the number of received beacons will vary.

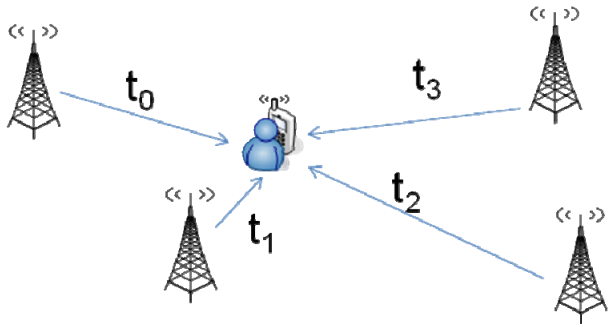


Figure 3.1: Message flow in device-based conventional positioning.

Since the link measurements are obtained directly in the MS from the beacons transmitted from the ANs, the only factor that attributes to the localization delay is the application location request interval μ_{loc}.

3.1.2 Measurement Collection for Centralized Cooperative Localization

In addition to MS-AN link measurements, the cooperative localization algorithms uses MS-MS ranging measurements and centralized computation of location estimates. In order to realize the collection of both types of measurements, as well as send back the location estimate to the MS, the message flow sketched in Figure 3.2 is used.

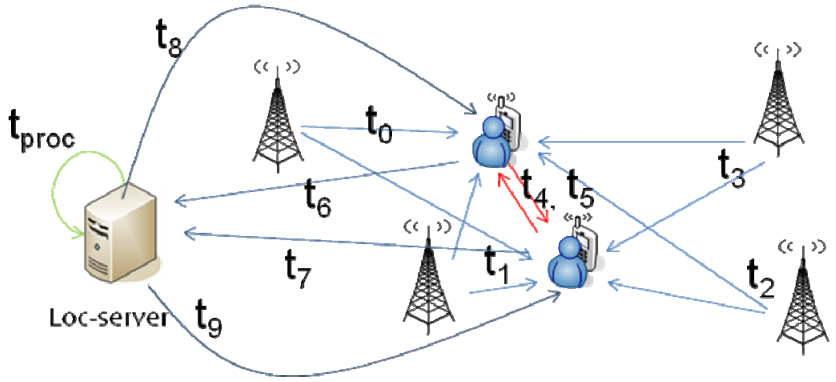


Figure 3.2: Message flow in centralized cooperative positioning.

In order to show the message flow more clearly, we consider the subflows individually in the following. Like the conventional algorithms, the cooperative algorithms rely on periodically transmitted beacons (every μ_{beacon} seconds) for MS-AN measurements. As before, we assume that the transmit power P_{tx} is fixed, known and equal for all ANs. Figure 3.3 shows how beacons transmitted from the ANs are first received and used for ranging at the MS. Hereafter a measurement packet, which contains the ranging measurement, is sent to the nearest AN and thereafter to the localization server, which is assumed to be connected to the AN by a wired infrastructure.

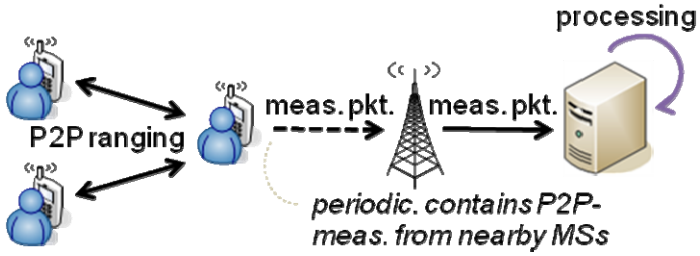


Figure 3.3: Message flow for beacon measurements in cooperative localization.

In addition to MS-AN measurements, the cooperative algorithms rely on MS-MS measurements. The flow of messages is shown in Figure 3.4. Whenever an MS senses another MS within d_{coop} meters, a P2P ranging measurement is made and sent to the localization server through the nearest AN. However, to reduce the amount of measurement packets being sent, P2P measurements are buffered and sent in a bundle every μ_{coop} seconds. As with the MS-AN measurements, the AN is assumed to be connected to the localization server by a wired infrastructure.

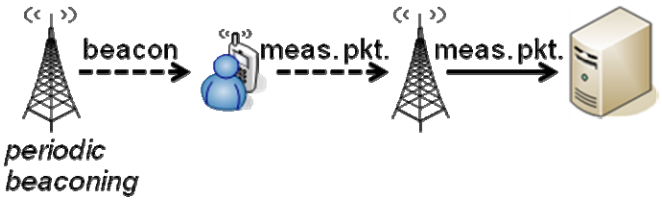


Figure 3.4: Message flow for P2P measurements in cooperative localization.

Having both MS-AN and MS-MS measurements at the localization server, we now need to provide the calculated position estimate to the MS. This is done by unicasting a message with the current location estimate of an MS to that MS, whenever a beacon from the nearest AN is received, as sketched in Figure 3.5.

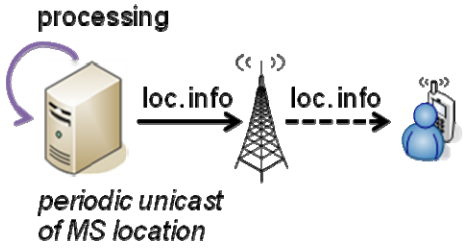


Figure 3.5: Message flow for location info message in cooperative localization.

The results in section 3.1.6 shows the delays and transferred bytes for each of the message collection protocols.

3.1.3 802.11a WiFi Network Model

The 802.11a WiFi network is simulated using ns-21 based on the mobility trace and the scenario specific parameters listed in Table 3.2. We use the 802.11ext module to simulate realistic 802.11a behavior. This ns-2 version includes a Nakagami fading model which has been parameterized according to Table 3.2 with model parameters $\Gamma = n$ and $m = \frac{(K+1)^2}{2K+1}$, where K is the Ricean K-factor, to approximate a Ricean fading environment.

Table 3.1 shows the sizes of the used messages. We have made the following assumptions regarding the used messages. The beacon is a standard 802.11 MAC frame, which follows the frame layout defined in [IEEE80211]. The beacon measurement is a 802.11 data frame with a payload consisting of the MAC id (6 bytes) of the AP and the estimated range (2 bytes). The P2P measurement bulk message size depends on the number of P2P neighbors in range. It is based on a data frame (28 bytes) where the payload is a 6 bytes MAC id and 2 bytes ranging value for each neighbor node. Finally, the location information message is a data frame with the node coordinates (x,y) encoded with 8 bytes each.

Message type	MPDU size (bytes)
802.11 MAC beacon	52
Beacon measurement	42
P2P measurement bulk	$28+(6+2)\cdot N_{\text{MSinrange}}$
Location information	44

Table 3.1: Message types

3.1.4 Group Mobility Model

A variation of the random waypoint that mimics group mobility is used in this work. In each group of nodes, one of the nodes acts as the reference node. For this node a waypoint and speed is chosen as usual for the random waypoint model (see [D45]). For the remaining nodes in the group the same speed is used and their waypoints are chosen, so that they are randomly placed within d_{spread} of the reference node's waypoint. An example of the resulting mobility tracks is shown in a $100x100m^2$ scenario. In this example there are 6 groups with 4 nodes in each group, shown with a unique color for each group.

[1]The ns-2 simulation is based on [CSJ+07], which has been updated with the author's patch from June 5th, 2009.

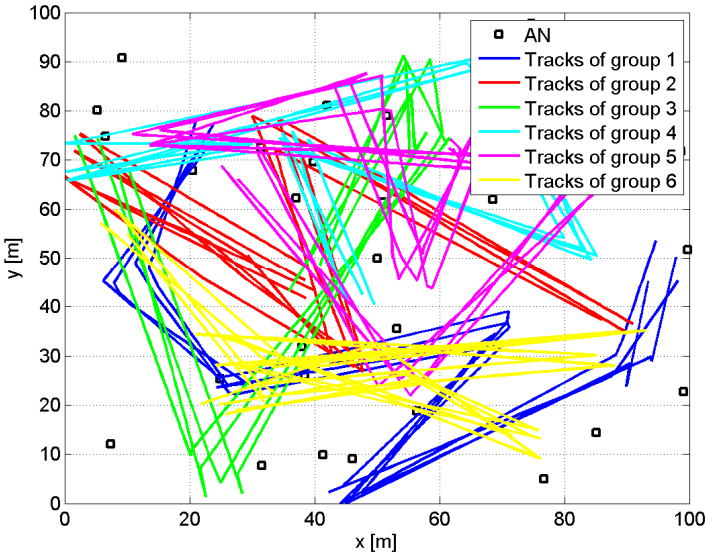


Figure 3.6: Group mobility simulation example (copy of Figure 2.56).

3.1.5 Evaluation Methodology

The considered localization algorithms have been evaluated with realistic communications constraints in a 4-step process as sketched in Figure 3.7.

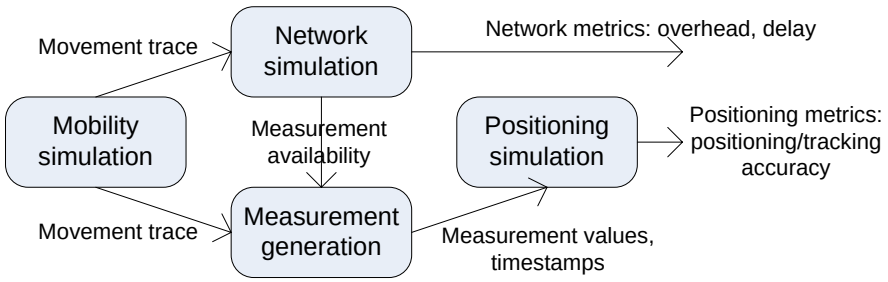


Figure 3.7: Simulation overview.

Initially, a common *Mobility simulation* is run, which results in a trace file that describes the AN positions and MS movements according to the random waypoint group mobility model described in the previous section. This mobility trace is then used as a basis for simulating the message collection protocols in the ns-2 based *Network simulation*. The output of this step is first the network-related performance metrics, and secondly this block also delivers a trace file specifying time stamps for when measurements are obtained and have been collected, according to the collection protocol. Using this trace file in combination with the mobility trace, the actual measurement values for the MS-AN and MS-MS links are being generated in the *Measurement generation* block using the models described in section 2.4. Finally, the *Positioning simulation* is run and positioning metrics are computed for the considered conventional and cooperative localization algorithms.

3.1.6 Simulation Results

The baseline simulation parameters are listed in Table 3.2.

Parameter	Value
Time	100 s
Size	$100x100m^2$
Number of ANs (N_{AN})	30
Number of MS groups (N_{groups})	6
Number of MSs per group ($N_{MS/group}$)	4
Max spread relative to ref. MS in group (d_{spread})	20 m

Movement speed ($\|\mathbf{v}\|$)	2 m/s
AN beacon interval (μ_{beacon})	1 s
P2P ranging interval (μ_{coop})	1 s
P2P ranging distance (d_{coop})	20 m
Location information update interval ($\mu_{\text{loc-info}}$)	1 s
MS application request interval (μ_{loc})	1 s
Localization processing time (μ_{proc})	0.1 s
Path loss exponent (n)	2.9
Rician K-factor (K)	6
Transmit power (P_{tx})	5 mW
802.11a PHY mode	6 Mbit/s, BPSK
Bandwidth	20 MHz
Frequency	5.18 GHz
Carrier Sense Threshold	−92 dBm
Noise floor	−106 dBm

Table 3.2: Network simulation parameters.

Initially we consider the effect of varying the transmit power. Figure 3.8 confirms that the number of ANs within carrier sense for each MS range increases with the transmit power, as expected. Notice that the curves for conventional and cooperative are (unsurprisingly) identical.

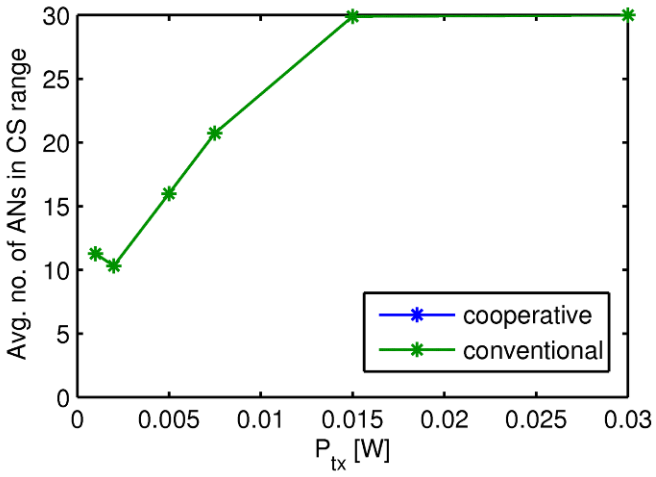


Figure 3.8: The number of ANs within carrier sense range for varying transmit power level.

In Figure 3.9 we show the average fraction of occupied channel time per AN. This metric is calculated by summing the time spent on transmissions within carrier sense range of each AN. The AN may overhear multiple simultaneous transmissions, since the considered $100x100m^2$ scenario does not constitute a single collision domain. In this plot it is clearly shown that the amount of occupied channel around each AN for the conventional measurement collection is much less than for the cooperative. Since the number of ANs and MSs is similar for conventional and cooperative, we can conclude that the conventional algorithm uses much less capacity for signaling, as we would expect. Further, all entities seem to be within the same collision domain for both $P_{\text{tx}} = 0.015$ and $P_{\text{tx}} = 0.030$ since the fraction of occupied channel does not change between these two parameter settings.

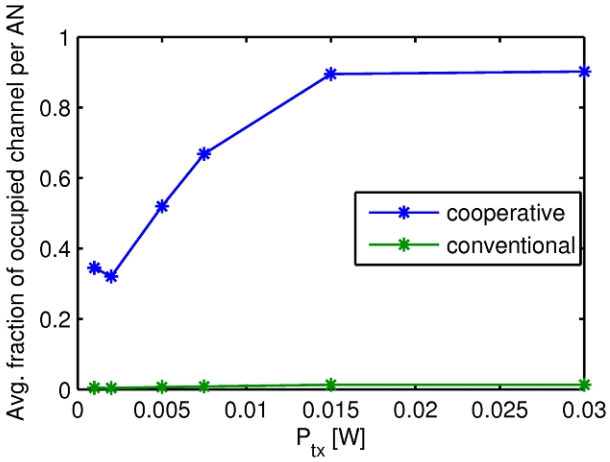


Figure 3.9: Average channel occupancy within carrier sense range for varying transmit power level.

We now consider the effect of varying the number of ANs. Figure 3.10 shows how the increasing number of ANs causes more traffic in the network. The average occupied channel around the AN can exceed 1 because not all entities are in the same collision domain. That is, multiple transmissions can be ongoing simultaneously if they are spatially well-separated [KLH+06].

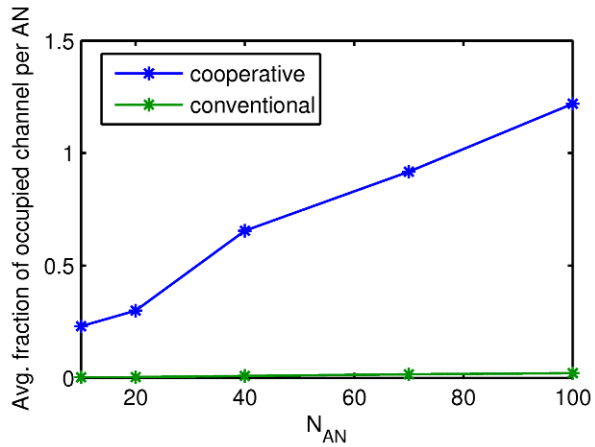


Figure 3.10: Average channel occupancy within carrier sense range for varying number of ANs.

Figure 3.11 shows the average localization delays for the conventional measurement collection and for the two types of measurements in the cooperative measurement collection. The localization delay is the time it takes from a measurement (received beacon or P2P ranging) is obtained at the MS, until the polling application on the MS has an updated location estimate. The delay for the conventional collection protocol does not change, since its delay only depends on the polling interval of the application μ_{loc}. On the other hand, the delay of the cooperative collection protocol seems to increase slightly with the increase of the number of ANs. If we look at Figure 3.10 we see that the channel occupation also increases with the number of ANs, thus the increase in delay may be due to a high level of contention among the network entities. On the other hand, the MS-MS measurements do not seem to be similarly affected by the increasing number of ANs. The reason for the MS-AN measurements being more sensitive to the number of ANs, could also be that many MSs receive the same beacon from an AN and followingly attempt to forward a beacon measurement at the same time. In case the level of contention is already high, the MSs must wait a considerable time to access the channel before the measurement can be delivered to the localization server.

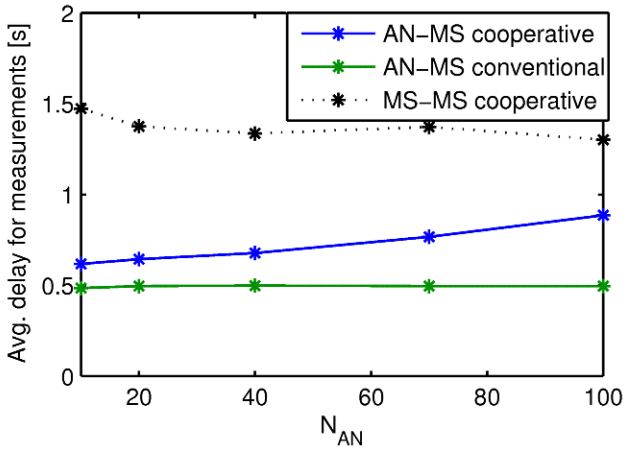


Figure 3.11: Average localization delay for varying number of ANs.

3.1.7 Conclusion

In this section we have described the realistic communication constraints for the conventional and cooperative localization algorithms considered in section 2.4 and evaluate network related performance metrics of the communication required for localization. Specifically, we consider the timing of measurement exchanges and availability of measurements through accurate simulations, and a group mobility model that mimics correlated user movements.

Measurements were obtained between ANs and MSs from medium range WiFi links and between cooperating MSs from short range UWB links. Measurements were collected to a processing entity through the WLAN infrastructure. It was shown that the introduction of realistic communications constraints resulted in an added delay, which is shown to have a significant effect on the positioning performance in section 2.4, especially for the cooperative algorithms. This delay is mainly due to the more complex measurement exchange that is necessary to realize the centralized cooperative positioning algorithms. In section 2.4, we also observed that increasing the number of cooperating MSs had a positive impact on the positioning performance, as expected due to added cooperation possibilities. However, this was only until a tipping point was reached and the performance became worse with additional cooperating MSs. This tipping point is likely a result of the communication overhead becoming large, which in turn leads to increased delays.

3.2 Extension of Prioritized and Decentralized Medium Access Schemes into Cooperative Tracking Scenarios

3.2.1 Motivations and Goals

In [D22], [DM09] and [Mensing10], a new scheduling technique was introduced to improve convergence speed and reduce overhead for jointly cooperative ranging and positioning algorithms supported by beacon-enabled Time Division Multiple Access (TDMA) protocols. The idea was to define priority levels or even to ignore irrelevant cooperative links while updating range measurements and location estimates in a distributed Least Squares approach based on asynchronous gradient descent. One of the described solutions considers discarding deliberately pair-wise range measurements and location updates to save time resources (i.e. guaranteed time slots) and reach faster positioning convergence under imposed peer-to-peer transactions, at the price of slight but consented average accuracy degradations. The solution relies on the prior analysis of the theoretical conditional positioning errors suffered at each mobile, following an approximated Cramer Rao Lower Bound (CRLB) analysis. A second family of “ordered” and “exhaustive” schemes, more adapted to aggregate and broadcast protocol schemes, tends to give priority to highly connected nodes. The underlying concept is that nodes with numerous neighbouring anchors can converge well and rapidly, and hence, that they can serve as reliable « virtual anchors ».

However, in spite of the significant gains observed on both overhead and convergence, experienced latency at network initialization still make the proposed solution mostly relevant in dense but static/quasi-static networks.

Besides, in [MDO08], prioritized medium access techniques relying on the “Dutch Auction” concept [RJ07] were adapted to high-precision real-time tracking applications. In comparison with classical TDMA-based schemes, the proposed uncoordinated and distributed prioritized solution appeared to be compliant with some critical non-cooperative tracking requirements in large-scale networks, under changing connectivity conditions with respect to anchor nodes. More particularly, this protocol tends to

favour high-speed targets as regards to the success rate (and hence, refreshment rate) of ranging packets issued at anchor nodes, as well as to the precision of resulting range measurements. However, this solution was uniquely intended at the ranging level in non-cooperative scenarios.

Based on the two previous contributions, we propose here to extend prioritized medium access protocol solutions and selective cooperation concepts, so that they can

- Benefit from selective and parsimonious cooperation
- Be evaluated at the location level under mobility (and not only at the ranging level), i.e. addressing cooperative tracking.
- Still be adapted to decentralized scenarios under partial and varying connectivity conditions.

Note that adapting existing protocol schemes from [MDO08] and [PULSII08] into the cooperative tracking problem implies:

- The definition of a new superframe structure that can support additional cooperative range measurements (i.e. not only with respect to anchors)
- The definition of a new adaptive local priority setting policy, i.e. not only to favour high-speed mobiles in terms of packets collision, but adequately responding to instantaneous and local needs in terms of cooperation.
- The design of an adequate decentralized filtering solution that can handle timely cooperative measurements under changing connectivity conditions and global nodes asynchronism.

In order to cope with the latter point, a sub-optimal (in the sense that location updates are performed asynchronously) but fully decentralized and cooperative extended Kalman filter has been proposed as well.

Overall, the expected benefits from all these adaptations are as follows:

- Enhance and homogenize location performances among mobile nodes, whatever their speed and vicinity/connectivity conditions
- Mitigate side issues that are traditionally inherent to cooperative/distributed approaches (e.g. location overhead with respect to communication means)

Note that the study hereafter mostly refers to the short-range cooperative links involved in the T1-A scenario under mobility (i.e. with no consideration with respect to fusion with cellular means). However, some of the presented concepts could be advantageously extended or adapted into dynamic heterogeneous scenarios as well (typically in the frame of WHERE2 WP2).

3.2.2 Prioritized Medium Access Schemes

3.2.2.1 Standard Non-Cooperative Scheme

Non-cooperative tracking algorithms necessitate ranging measurements, at least with respect to anchors nodes. In the following, we consider that each of those measurements between two nodes is based on TOA estimation and the exchange of 3 packets, including a ranging request, a ranging response, and an acknowledgement that is also used for relative clock drift estimation and compensation. Moreover, we aim at delivering ranging results at mobile nodes so that the location computation can be decentralized.
In the standard non-cooperative prioritized protocol scheme, the idea is that each target can locally set a priority level for the transmission of ranging responses back to anchor nodes. This level is then compared with a common count down (aligned on the superframe timeline) to determine the position in time of the transmission attempt.

The example shown on Figure 3.12 (a) illustrates the periodical superframe structure and timeline. The complete procedure includes:

- Broadcasted requests and broadcasted acknowledgements from anchors to mobiles, in pre-convinced slots respectively at the beginning and the end of each superframe
- A common silent count-down (with no stringent constraints on synchronization among the nodes, but time requirements similar to that of classical TDMA slotted communications with beacon-aided piconet synchronization)
- Local priority level at each target, depending on estimated local speed
- A broadcasted response from a target (attempt) once the current count-down is below the local priority level
- An inactive period at the end of the superframe

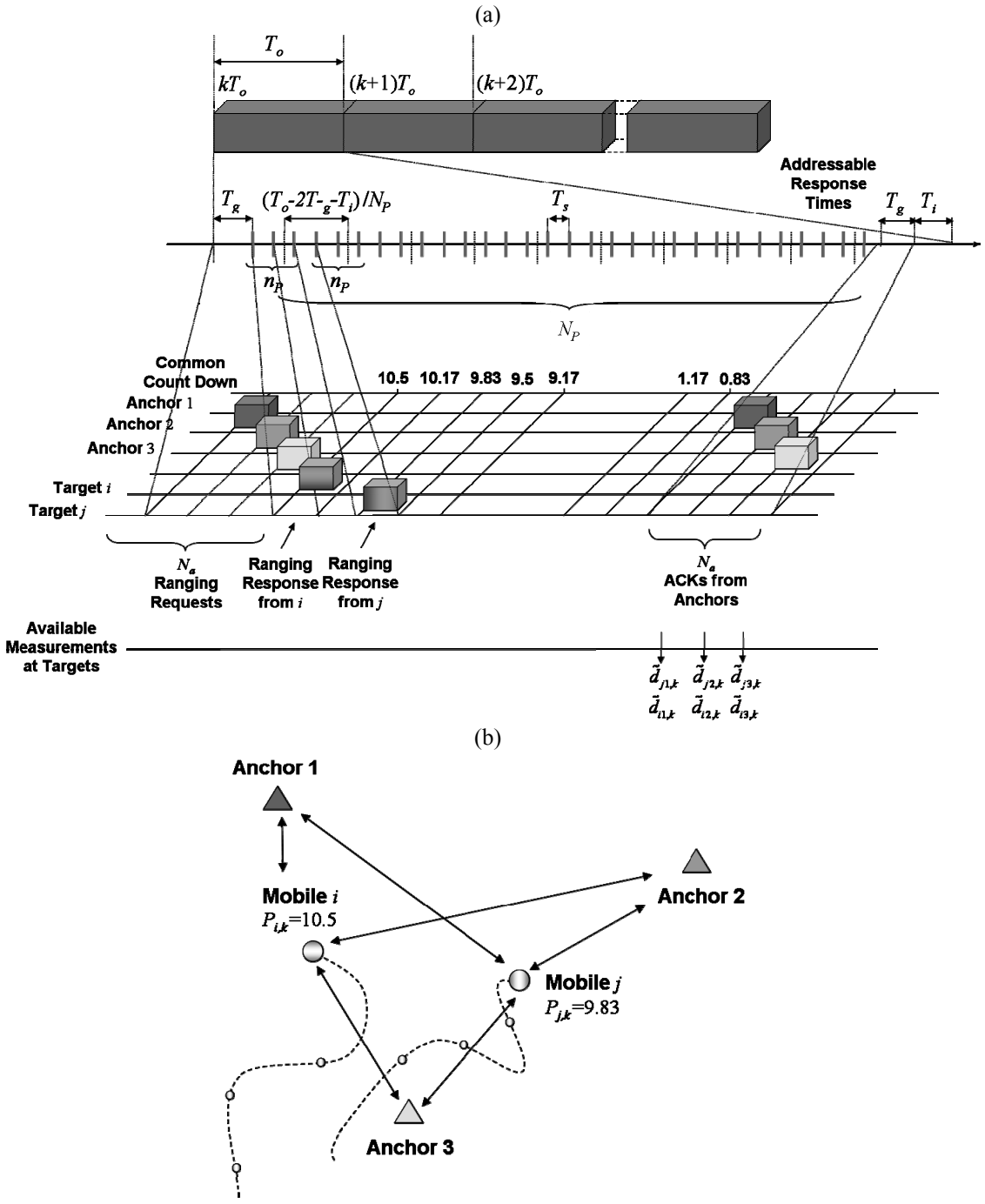


Figure 3.12 : Example of superframe structure in the non-cooperative decentralized and prioritized access scheme (a) and related tracking scenario (b).

More precisely, during the response period, competing mobiles have to decide in which slots they can transmit their responses, based on the definition of local priority levels. For instance, one intuitive priority setting relies on the experienced module of the 2D speed or traveled distance between consecutive updates of the target positions. In the nominal embodiment, depicted as BRR-BA (for *Broadcasted Ranging Requests – Broadcasted Answers*), the index of the response slot corresponds to the priority level $P_{i,k}$ at each mobile node i for the k^{th} period (superframe), taken among $N_R = N_p n_p = Cst$ addressable positions, on an equally-spaced priority grid, as follows:

$$P_{i,k} = C_{i,k} n_p + f_{i,k} = \left\lfloor N_p \hat{V}_{i,k} \Big/ \left(s V_{\max} \right) \right\rfloor n_p + f_{i,k} \tag{3.1}$$

where $C_{i,k}$ is a coarse priority level taken among N_p possible values, as a deterministic function of the estimated speed $\hat{V}_{i,k}$ (that can be only a coarse estimate), s a scaling factor (with respect to the priority grid), and $f_{i,k}$ is a discrete random fine priority level taken among n_p values with:

$$p\left(f_{i,k}\right) = \frac{1}{n_p}\sum_{l=1}^{n_p}\delta\left(f_{i,k} - l\right) \tag{3.2}$$

Note that the previous scheme tends to favour high-speed nodes in terms of response packet collisions for practical speed distributions with more frequent low-speed nodes (e.g. like the Rayleigh distribution considered in [MDO08] and [PULSII08], resulting from the centered Gaussian 2D speed terms affecting Cartesian coordinates). However, note that without loss of generality, the deterministic function $C_{i,k}$ can be adjusted to any kind of priori speed distribution.

At this point, two major enhancements to the basic scheme were also provided to:

- Be less sensitive to the initial underlying speed distribution, getting adapted to the instantaneous "capacity/load" of neighboring anchors (in terms of colliding packets received from competing mobiles) and adaptively changing at the target the number of fine priority levels to account for such anchors activity
- Consider dynamic election of coarse priority levels locally, depending also on the history of experienced past collisions / attempts at each target (but not only on the experienced speed)

Overall, based on this enhanced scheme, it has already been shown (in canonical scenarios) that the slowest nodes of a network would benefit from less frequent ranging updates with respect to a few anchors whereas high-speed nodes could successfully enjoy frequent ranging updates with respect to numerous anchors.
Further precisions can be found in [PULSII08]. In the following, we will consider this prioritized access scheme as the nominal "Fully Decentralized but Non-Cooperative" scheme (FDNC).

3.2.2.2 New Cooperative Access Schemes

We propose to modify the structure of the superframe and the priority rules so that new cooperative links can be incorporated in the tracking problem.
One first challenge is that the allocation of new time resources should still be made on a decentralized basis, enabling mobile-centric tracking over large-scale networks with limited connectivity and avoiding the relay of intermediary information to a sink (required by centralized computations). Another stake is that a parsimonious use of cooperative links and measurements is preferable to fulfil real-time and energy-efficiency requirements, hence necessitating cooperation rules that depend mostly on the local and instantaneous needs expressed at mobile nodes.
The idea consists in extending the period dedicated to ranging requests at the beginning of each superframe (addressed uniquely by anchors in the non-cooperative case so far) so that only the mobiles that can consider themselves as relevant cooperative nodes (with respect to other mobiles), start behaving as virtual anchors and send ranging requests as well. Obviously the same mobiles can also send acknowledgements back to other mobiles at the end of the superframe to complete cooperative multiple-way ranging transactions with clock drift estimation and compensation, like in [MDO08]. These two period extensions do not have an important impact on the best achievable refreshment period. In fact, in usual cases, this extra-time can be compensated by a judiciously adjusted inactive period.
Beyond this generic principle, one remaining question is about the instant (i.e. in which superframe and in which slot) when these cooperative nodes should send their request packets and serve as virtual anchors. Three selective cooperation schemes are investigated and compared hereafter in terms of location precision and slot occupancy.

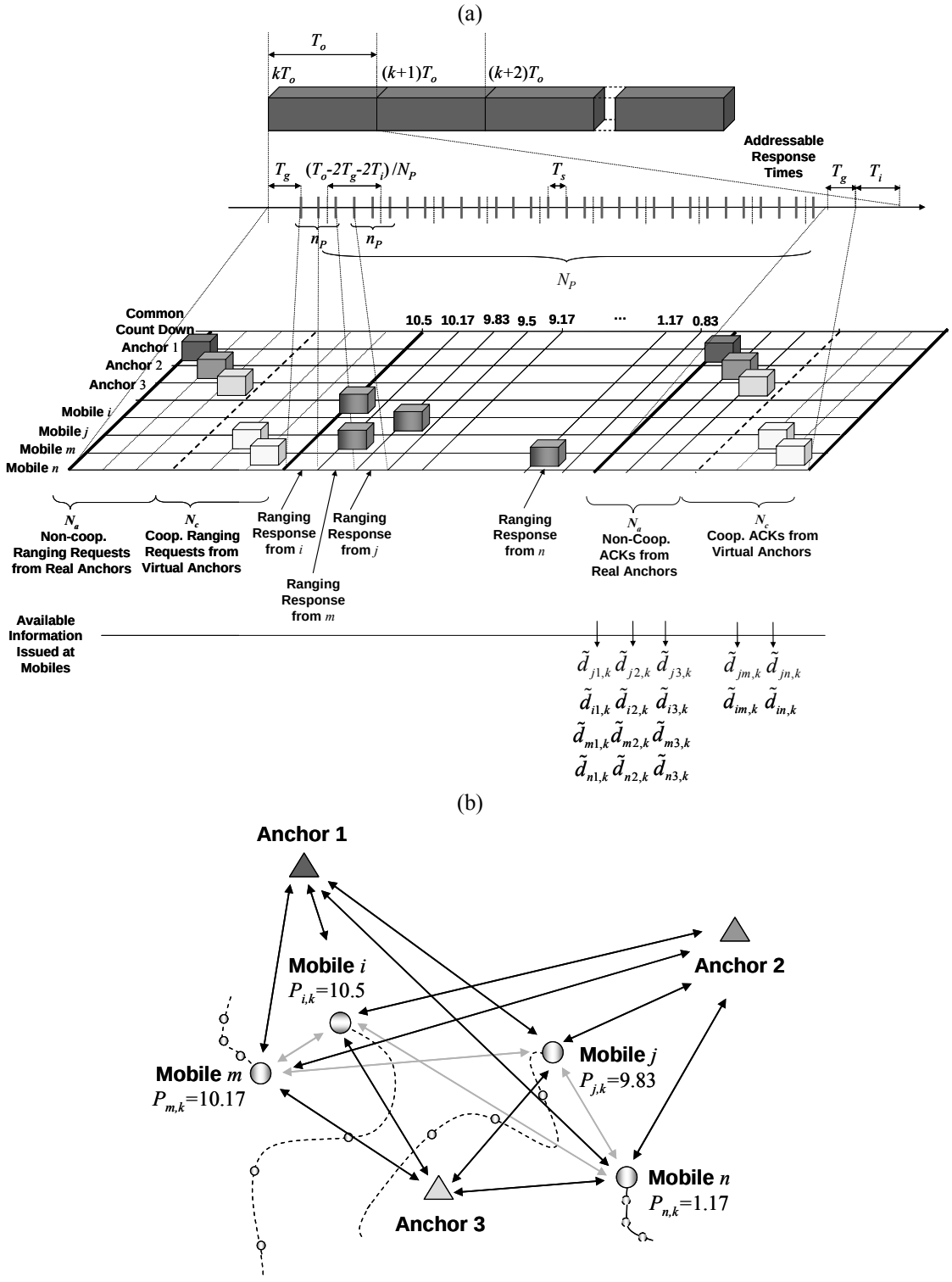


Figure 3.13 : Example of superframe structure in the cooperative decentralized and prioritized access scheme (a) and related tracking scenario (b).

Fully Centralized and Exhaustive Cooperation

The first scheme is mostly proposed for comparison purposes. It consists in pre-allocating the new N_c slots available in the cooperative request period (See Figure 3.13 (a)) to all the nodes sequentially (e.g. following a Round-Robin scheme), so that each mobile is ensured to serve as a virtual anchor with respect to other neighbouring mobiles after a certain amount of time, with no other competing transmissions and regardless of its actual relevance or legitimacy. As an example, in a network comprising 100 mobiles and with N_c=5, each mobile would send one cooperative ranging request once every 20 superframes. This "Fully Centralized and Exhaustive Cooperative" (FCEC) access is energetically inefficient in the sense all the N_c slots are occupied. It is also subject to counterproductive links choices in terms of cooperation, from both reliability and timeliness points of view. Finally, such cooperative access can hardly support

dynamic network topologies (i.e. involving a varying number of nodes, with newly associated or withdrawn nodes).
For the allocation of the cooperative request period slots, the existence of a physical coordinator or the exchange of control packets, e.g. for the confirmation of guaranteed time slot allocation, are not essential. As an example, in each superframe, one device can evaluate its attributed cooperative request slot depending on the current superframe number and its address (attributed during the association procedure). Hence, the network only has to define its maximal capacity N_{tot} in terms of devices number. The N_c addresses can be chosen as follows:

$$Address \in \left[\left((k-1) \times N_C \right)_{[N_{tot}]} \quad \left(k \times N_C \right)_{[N_{tot}]} \right] \tag{3.3}$$

where k is the current superframe index and $\left(A \right)_{[B]}$ is A modulo B.

Half-Centralized and Selective Cooperation
The second scheme is the same as the previous one from an allocation perspective, but the decision to send requests (or not) in cooperative slots is decentralized and up to each mobile, depending on its assumed reliability. As an example, if the covariance of the latest estimated location locally exceeds a certain threshold T set a priori, then one mobile can decide to keep silent in its assigned slot. On the contrary, if the so-called location uncertainty looks more favourable, this mobile can decide on its own to address the assigned slot with a cooperative ranging request. One advantage in comparison with the initial fully centralized scheme is that this selective transmission of cooperative requests is expected to prevent from irrelevant and useless cooperative links. Note that this concept is rather similar to that of the previous study on static positioning [D22], where deliberate link discarding was applied based on local performance expectation. This access scheme is also more efficient in terms of energy consumption, in the sense useless transmissions are avoided. However, since each of the corresponding empty slots is pre-allocated specifically for one node, no benefits can be expected in terms of achievable refreshment rates or data rates at the application level (unless slot reuse is applied for other purposes than tracking). Hence, another problem is that this access scheme still necessitates prior centralized allocation and that resources can be spoilt for nothing (in case allocated slots are not addressed). This "Half-Centralized and Selective Cooperative" (HCSC) scheme is however useful to verify the amount of energy that could be saved (in terms of lower duty cycle transmission activity), at the price of slight but controlled location degradations (resulting from a lack of cooperation in comparison with the exhaustive case). Finally, it also enables to illustrate the operating trade-off for the selection threshold T. Indeed, if the latter is too low, just a few mobiles are turned into anchor nodes (and the level of cooperation is low accordingly), whereas if too high, the access scheme is too permissive and reduces to the exhaustive FCEC case.

Fully Decentralized and Selective Cooperation
According to this third scheme, there is no prior centralized allocation of cooperative slots, but the decision when to send a cooperative ranging request (for both the superframe and the cooperative slot) is made locally at each mobile, following a probabilistic approach. Accordingly, cooperative slots are no more guaranteed any more, but collisions are tolerated between competing virtual anchors.

First, at each superframe k, each mobile node determines a probability to send a cooperative ranging request, as a function of its assumed reliability, e.g. based on the covariance of its estimated location (e.g. as one of the tracking filter outputs):

$$P_{VA}^{k} = \exp\left(-\frac{\left| C \right|^{k-1}}{c_s} \right) \tag{3.4}$$

where c_s is a scaling factor reflecting the strength of the selective process (e.g. equal to 0.03 in the following), and $\left| C \right|^{k-1}$ an indicator on reliability of the latest location estimate available at time $k-1$.
Accordingly, during the current superframe, the actual Virtual Anchor (VA) status is treated as a realization w_{VA}^{k} of a bi-state discrete random variable $W_{VA}^{k} = \{0,1\}$, with probability values $\left\{ 1-P_{VA}^{k}, P_{VA}^{k} \right\}$.
If $w_{VA}^{k} = 0$, the node does not send any request packet. On the contrary, if $w_{VA}^{k} = 1$, it has to choose one slot among N_c. In the following, we consider a simple uniformly distributed choice, like for the fine priority levels defined in the standard ranging response period.
As already mentioned, this "Fully Centralized and Selective Cooperation" (FCSC) access scheme is subject to collisions, which in turn might alter cooperation benefits in comparison with HCSC (e.g. in case of high setting for c leading to high probability values P_{VA}^{k} or with under dimensioned N_c). But on the other hand, one can expect a better usage of empty slots and a spatial reuse of the available resources, authorizing parallel cooperative updates for links geographically located in distant non-interfering

network areas. Hence, in average, relevant updates are expected to occur even more rapidly than in the previous HCSC case, so that location performance can be enhanced.

3.2.3 Decentralized Cooperative Tracking Filter

In order to exploit all the TOA-based ranging measurements made available with the new cooperative medium access schemes, a decentralized tracking filter is required. In comparison with the centralized cooperative filter, we consider a sub-optimal embodiment that only handles the update of the recipient node location and speeds at each superframe *k*, while the neighbouring mobiles provide the recipient with their latest estimated coordinates, speeds, and estimation covariance obtained at time *k*-1.

3.2.3.1 State Vector and Equation

The state vector associated with one mobile m_i is as follows:

$$S_i^k = \left[x_i^k, y_i^k, v_{x_i}^k, v_{y_i}^k \right] \tag{3.5}$$

where $\left\{ x_i^k, y_i^k \right\}$ are the 2D Cartesian coordinates, $\left\{ v_{x_i}^k, v_{y_i}^k \right\}$ are the 2D velocities.

The related state equation is as follows:

$$S_i^k = FS_i^{k-1} + N_i^k \tag{3.6}$$

where $N_i^k = \left[n_{x_i}^k, n_{y_i}^k, n_{vx_i}^k, n_{vy_i}^k \right]$ is a centred Gaussian vector and

$F = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ is a simple transformation matrix accounting for a simple linear mobility model (at least between bounces on the scene borders), with *dt* corresponding to the best refreshment period (i.e. equivalent to a superframe duration T_0 in our case, as shown on Figure 3.13). This piecewise linear model is one of the models investigated and discussed in WP4 T4.2 (e.g. see [D45]).

3.2.3.2 Observation Vector and Equation

The considered observation vector $N_a^k + N_c^k$ comprises terms as follows:

$$\begin{aligned} \tilde{Z}_i^k &= \left[\tilde{d}_{m_i a_1}^k \;\; .. \;\; \tilde{d}_{m_i a_l}^k \;\; .. \;\; \tilde{d}_{m_i a_{N_a^k}}^k \;\; \tilde{d}_{m_i m_1}^k \;\; .. \;\; \tilde{d}_{m_i m_j}^k \;\; .. \;\; \tilde{d}_{m_i m_{N_c^k}}^k \right] \\ &= \left[d_{m_i a_1}^k \;\; .. \;\; d_{m_i a_l}^k \;\; .. \;\; d_{m_i a_{N_a^k}}^k \;\; d_{m_i m_1}^k \;\; .. \;\; d_{m_i m_j}^k \;\; .. \;\; d_{m_i m_{N_c^k}}^k \right] + \eta_i^k \\ &= G_i^k \left(S_i^k \right) + \eta_i^k \end{aligned} \tag{3.7}$$

where $\left\{ m_j \right\}, j = 1..N_c^k$ and $\left\{ a_l \right\}, l = 1..N_a^k$ refer to successful neighbouring mobiles viewed as virtual anchors and neighbouring anchors respectively (i.e. complete ranging transactions have been issued into m_i at the end of the *k-th* superframe with respect to these nodes), $d_{m_i a_l}^k = \sqrt{\left(x_{m_i}^k - x_{a_l}^k \right)^2 + \left(y_{m_i}^k - y_{a_l}^k \right)^2}$ is the true distance at time *k* between mobile m_i and anchor a_l, $d_{m_i m_j}^k = \sqrt{\left(x_{m_i}^k - x_{m_j}^k \right)^2 + \left(y_{m_i}^k - y_{m_j}^k \right)^2}$ is the true distance at time *k* between mobiles m_i and m_j, $G_i^k(.)$ is a non-linear function of the current state accounting for all the previous distances, and η_i^k is assumed to be a centred Gaussian noise vector which standard deviation terms are compliant with [MDO08]:

$$\sigma = \frac{c\sigma_{TOA}}{2} \sqrt{\frac{2T_T}{T_T + T_A} \left(\frac{T_T}{T_T + T_A} - 1 \right)} \tag{3.8}$$

where T_T is the response time at the updated mobile (i.e. the time elapsed between the reception of the ranging request from the anchor or the virtual anchor and the emission of the ranging response), T_A is the response time at the anchor or the virtual anchor (i.e. the time elapsed between the reception of the ranging response and the emission of the acknowledgement), σ_{TOA} is the standard deviation of unitary TOA estimates (associated with the 3 received packets involved in the complete ranging procedure), and c the speed of light.

Accordingly, depending on the respective transmission slots adopted by two devices over one link (cooperative or not), the standard deviation associated with the current range measurement is specific and perfectly predictable as a function of basic protocol durations and expected TOA uncertainty (typically the slot duration T_S here).

3.2.3.3 Overall Filter Structure

At the *k-th* superframe, each mobile node m_i locally updates its own estimated location and speed, based on:

- Its previous estimates $\hat{S}_i^{k-1}$
- The set of N_c^k successful cooperative range measurements with respect to neighbouring mobiles serving as virtual anchors, i.e. $\left\{\tilde{d}_{m_i m_j}^k\right\}, j=1..N_c^k$
- The set of N_c^k estimated locations and speeds associated with neighbouring mobiles serving as virtual anchors (information transmitted in the ranging packets issued at m_i during the *k-th* superframe), i.e. $\left\{\hat{S}_j^{k-1}\right\}, j=1..N_c^k$, along with the related covariance matrices $\left\{C_j^{k-1}\right\}, j=1..N_c^k$
- The set of N_a^k successful non-cooperative range measurements with respect to neighbouring anchors $\left\{\tilde{d}_{m_i a_l}^k\right\}, l=1..N_a^k$

Overall, the filter synopsis and structure is inspired by a standard Extended Kalman Filter (EKF), as follows (for N_m mobiles over K successive superframes):

For $k=1..K$

For $i=1..N_m$

$$\hat{S}_i^{k/k-1}=F\hat{S}_i^{k-1}$$
$$C_i^{k/k-1}=FC_i^{k-1}F^T+Q_i^{k-1}$$

For $j=1..N_c^k$

$$\hat{S}_j^{k/k-1}=F\hat{S}_j^{k-1}$$
$$C_j^{k/k-1}=FC_j^{k-1}F^T+Q_j^{k-1}$$

end

$$T_i^k=g_i^k C_i^{k/k-1}\left(g_i^k\right)^T+g_{i,neigh}^k C_{i,neigh}^{k/k-1}\left(g_{i,neigh}^k\right)^T+R_i^k$$
$$K_i^k=C_i^{k/k-1}\left(g_i^k\right)^T\left(T_i^k\right)^{-1}$$
$$\hat{S}_i^k=\hat{S}_i^{k/k-1}+K_i^k\left(\tilde{Z}_i^k-G\left(\hat{S}_i^{k/k-1}\right)\right)$$
$$C_i^k=C_i^{k/k-1}-K_i^k T_i^k\left(K_i^k\right)^T$$

end

end

(3.9)

where $g_i^k=\partial G_i^k\left(\hat{S}_i^{k/k-1},\hat{S}_{i,neigh}^{k/k-1}\right)\Big/\partial\hat{S}_i^{k/k-1}$ is a $\left(N_a^k+N_c^k\right)\times 4$ a Jacobian matrix comprising derivatives with respect to m_i 's state variables, evaluated at the predicted state and $g_{i,neigh}^k=\partial G_i^k\left(\hat{S}_i^{k/k-1},\hat{S}_{i,neigh}^{k/k-1}\right)\Big/\partial\hat{S}_{i,neigh}^{k/k-1}$ a $\left(N_a^k+N_c^k\right)\times 4\left(N_a^k+N_c^k\right)$ Jacobian matrix comprising derivatives with respect to neighbours' 2D coordinates and speeds (if any neighbour is available), evaluated at the predicted state as well, with $\hat{S}_{i,neigh}^k=[S_{a1}\quad ...\quad S_{al}\quad ...\quad S_{a_{N_a^k}}\quad \hat{S}_{m_1}^{k/k-1}\quad ...\quad \hat{S}_{m_j}^{k/k-1}\quad ...\quad \hat{S}_{m_{N_c^k}}^{k/k-1}]$,

$$C_{i,neigh}^{k/k-1} = \begin{bmatrix} [0]_{4\times4} & & & & & & & \\ & \ddots & & & & & & \\ & & [0]_{4\times4} & & & 0 & & \\ & & & \ddots & & & & \\ & & & & [0]_{4\times4} & & & \\ & & & & & C_{m1}^{k/k-1} & & \\ & & 0 & & & & \ddots & \\ & & & & & & & C_{mj}^{k/k-1} \\ & & & & & & & & \ddots \\ & & & & & & & & & C_{m_{N_c^k}}^{k/k-1} \end{bmatrix} \quad (3.10)$$

is a $4\left(N_a^k + N_c^k\right)\times 4\left(N_a^k + N_c^k\right)$ block-diagonal covariance matrix associated with m_i 's neighbours estimates and finally,

$$R_i^k = \begin{bmatrix} \sigma^2_{m_i a_1} & & & & & & & \\ & \ddots & & & & & & \\ & & \sigma^2_{m_i a_l} & & & 0 & & \\ & & & \ddots & & & & \\ & & & & \sigma^2_{m_i a_{N_a^k}} & & & \\ & & & & & \sigma^2_{m_i m_1} & & \\ & & 0 & & & & \ddots & \\ & & & & & & & \sigma^2_{m_i m_j} \\ & & & & & & & & \ddots \\ & & & & & & & & & \sigma^2_{m_i m_{N_c^k}} \end{bmatrix} \quad (3.11)$$

is a $\left(N_a^k + N_c^k\right)\times\left(N_a^k + N_c^k\right)$ diagonal matrix associated with observations, containing ranging measurement variances.

In comparison with centralized cooperative filters (e.g. See 2.4.2.2), the solution can be viewed as sub-optimal in the sense all the nodes locations are not estimated simultaneously/synchronously, knowing all the cooperative measurements at each superframe. In a decentralized and large-scale context, one idea is to respect the global asynchronism issue and the specific connectivity conditions observed at each node. One second aspect is to consider that the update of cooperating neighbours would imply further latency (to transmit back estimation results) and would impose to maintain a live neighbours table in each mobile under mobility. Consequently, we just make sure that each mobile is only updated once per superframe.

It might happen that no observation is available (i.e. no range measurement is issue) for a superframe in certain nodes. In this case, we simply use the prediction step as the final estimation one (i.e. with no filter correction).

In the next superframe *k+1*, the mobile m_i can encapsulate the information $\hat{S}_i^k$ and C_i^k in its ranging packets directed to other mobiles (uniquely in the ranging response if staying a simple mobile, i.e. $w_{i,VA}^{k+1} = 0$ or in the ranging request and response if becoming a virtual anchor, i.e. $w_{i,VA}^{k+1} = 1$).

3.2.4 Results

In this section, we provide a few illustrating simulation results to point out the main advantages and drawbacks of the protocol schemes introduced previously.

3.2.4.1 Simulation Parameters

In order to evaluate and enhance the performance of the proposed protocol under mobility, a packet-oriented event-driven simulator has been specifically adapted. In this simulator, since the transmission range is evaluated as a function of the actual distance and instantaneous SINR when receiving response packets at anchors, realistic collision conditions and partial connectivity can be adequately accounted under mobility.

In addition, performing ranging transactions in our simulator, each device (anchor or mobile) utilizes its own clock for which the perceived local time (i.e. read on the local clock) is represented as a function of the absolute time, the relative frequency offset and clock shift (both with respect to an ideal clock providing the absolute time). Hence, post-corrections are applied according to the description made in [MDO08], so that plausible range measurements can be generated overall, depending on both the protocol durations and the quality of unitary TOA estimates for each of the 3 received packets of a complete ranging transaction, as previously pointed out. In our case, each estimated TOA affected by centred Gaussian uncertainty with a standard deviation $\sigma_{TOA} = \sqrt{Kd^{\alpha}}/c$ as a function of the distance d, with $K = 0.001$ and $\alpha = 2$, like for the unbiased LOS channel configurations in [DM09] (e.g. $\sigma_{TOA} \approx 2ns$ at $d = 20m$). These dispersion figures are compliant with the experimental validations made in typical indoor environments with real WP5 Impulse Radio Ultra Widdeband platforms (e.g. see [D53]). However, for the sake of simplicity and since the focus is mostly put here on medium acess and cooperation, we do not consider NLOS biased measurements in our filter design and simulation results. Note however that the study could be extended in NLOS configurations in future investigations (i.e. coupling our cooperative access scheme and decentralized filter solutions with some bias detection/estimation/mitigation techniques investigated in WHERE WP2 T2.1 [D23] and WP4 [D46]).

Moreover, each mobile chooses a 2D random speed satisfying $V_{i,k} < V_{\max}$, each speed component (i.e. along x and y coordinates) following a centered Gaussian distribution with standard deviation $\sigma_V \approx V_{\max}/3$ so that $V_{i,k}$ follows a Rayleigh distribution with a standard deviation $\sigma_V \approx V_{\max}/3$. In our simulations, the maximum tolerated speed is $V_{\max} = 5m.s^{-1}$ accordingly.

As shown on Figure 3.14, the network of interest is composed of N_m =91 cooperative mobile nodes deployed in a 40m×40m area. Depending on connectivity conditions (transmission range, interferences and collisions….) these mobiles can communicate with up to 8 surrounding anchors and 1 central anchor, as well as with other moving mobiles. The 91 mobiles are randomly drawn on the scene, with a random constant speed following the above mentioned Rayleigh distribution. Moreover, a bounce-oriented mobility model is also assumed. Accordingly, whenever a mobile reaches the border of the scene, the trajectory is reflected back on the scene. This tends to create discontinuities in the generated trajectories, which are not taken into account in the filter. This kind of mobility patterns has been more deeply investigated and discussed in WP4 T4.2 (See [D45]).

As regards to the initial guess $\left\{\hat{S}_i^0\right\}, i = 1..N_m$, two cases are considered: 1) so-called “perfect” initialization with exact knowledge of coordinates but random speeds and 2) “random” initialization with both random coordinates and speeds. In both options, the initial estimation covariance reflects corresponding uncertainty. In the following, on the one hand we compare the centralized schemes (i.e. FSEC and HCSC) with a basic non-cooperative scheme. On the other hand, we compare the fully distributed scheme (FDSC) with the non-cooperative scheme. Consequently, the tracking gain from incorporating additional cooperative ranging transactions is fairly assessed under the same collision conditions and maximum refreshment rates (in order to submit the algorithms to exactly the same random occurrences). In all the simulations, we consider nominal superframe duration of 300ms.

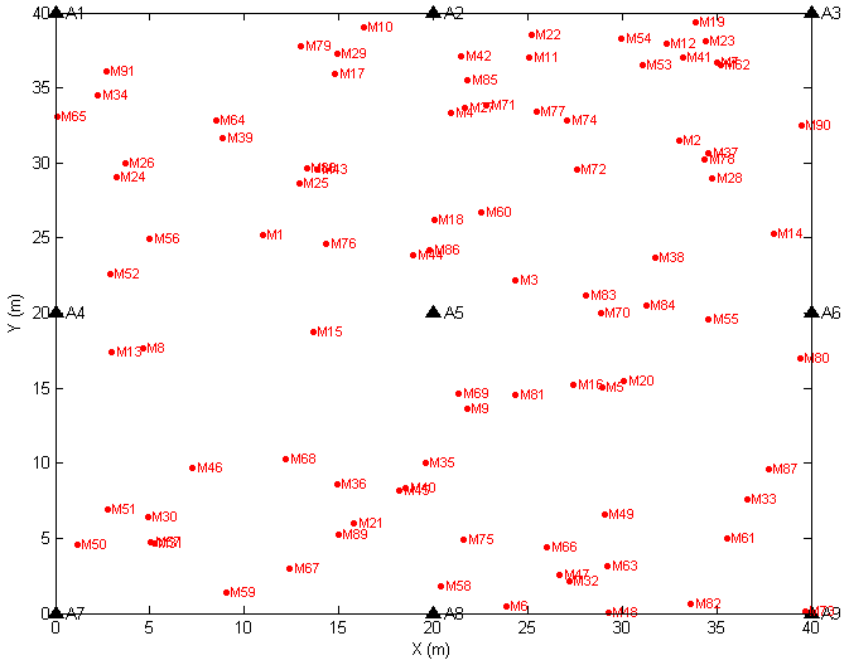


Figure 3.14 : Example of network topology with 9 regularly spaced anchors (black triangles) and 91 mobiles (red dots).

3.2.4.2 Simulation Results

First of all, tracking performances are characterized here in terms of global network error (e.g. the average location error over all the mobiles devices at a given superframe), for different centralized schemes under the same conditions in the response period (i.e. collisions, priority setting and refreshment rate), we simulate a common scenario in terms of mobility and measurement noises for three different Extended Kalman Filters:

- FCEC filter: We consider the whole observation vector, i.e. the successful ranging with respect to each neighbouring anchors or virtual anchors.
- HCSC filter: We consider a limited observation vector composed of the successful ranging measurements with respect to neighbouring anchors and virtual anchors with the latest estimation covariance inferior to a certain threshold *T*. Obviously, if this scheme is selected, it will be more efficient in terms of energy consumption to take the decision in the transmitter than in the receiver.
- NC filter: We consider only the observations with respect to the neighbouring anchors.

We can also recall that the cooperative ranging transactions realized with respect to virtual anchors do not have any impact on the local priority setting (central period dedicated to mobile responses).
Figure 3.15 shows an example of global location error behaviour (averaged over all the mobiles), as a function of the elapsed time for centralized schemes (both non-cooperative and cooperative) and under the same collision conditions. A varying selection threshold is applied for HCSC, illustrating gradual/intermediate cooperation effects and corresponding error gains, between FCEC and NC. In the following, the selection covariance threshold of the HCSC scheme is set empirically to 0.5m², as a good trade-off preserving both location performances and energy savings.
We can note that with a selection threshold T=0.05, the performances are similar to that of the non-cooperative scheme, because the number of cooperating devices is too low, whereas with a threshold T=0.5, performances are satisfactory because the number of cooperating devices is larger. Note that the selection of cooperative devices could be based on the observation of the dynamic estimation covariance CDF, as a function of time.

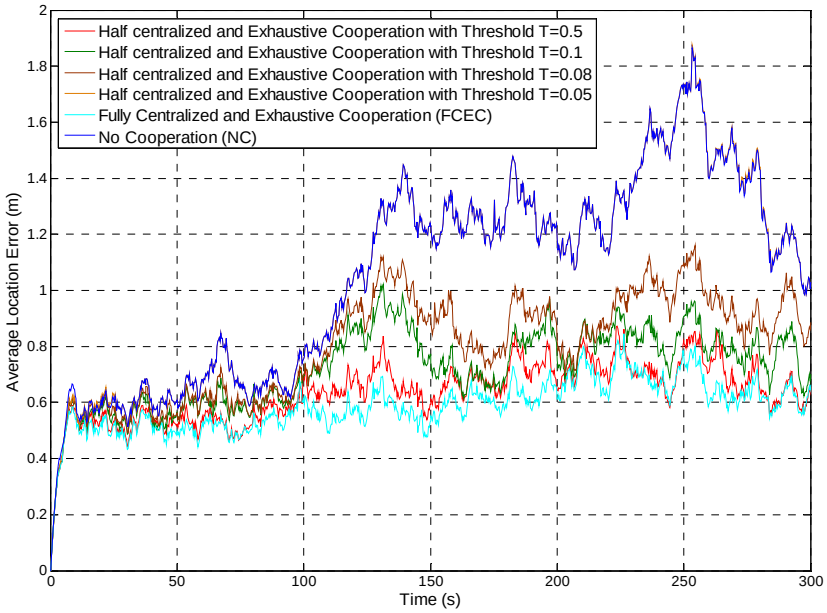


Figure 3.15: Example of global average location error (over all the mobile nodes), as a function of the elapsed time and HCSC selection threshold values and with perfect initialization.

Figure 3.16 displays the CDF of the location error for the centralized schemes at a given time t=60s (i.e. after 200 superframe). In this example, 95% of the mobile devices can achieve a location error lower than 1.33m (resp. 1.36 and 1.94) for HCSC (resp. FCEC and NC).

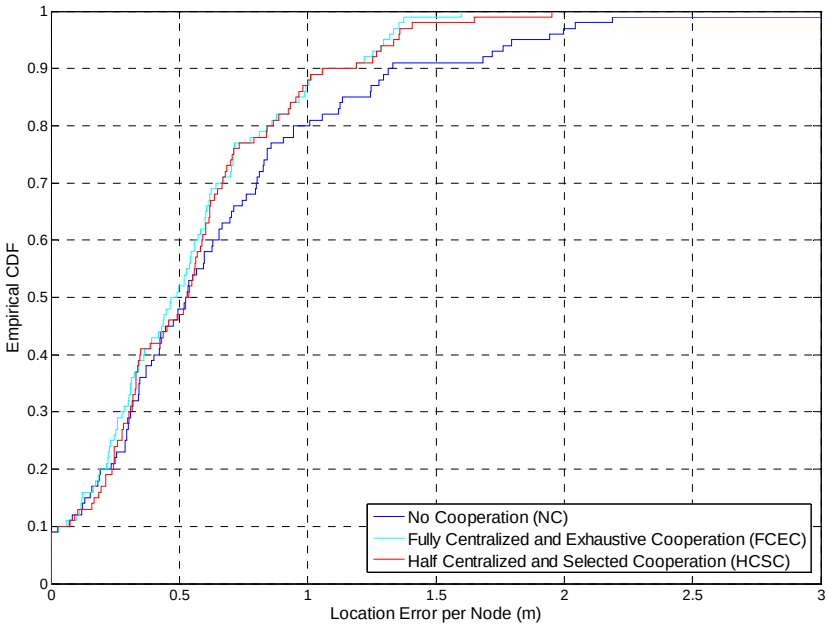


Figure 3.16: CDF of location errors after 200 Superframes for centralized cooperative and non-cooperative schemes.

Another interesting result is the distribution of the location error per node as a function of the estimated location covariance, at a given time (e.g. t=60s, i.e. after 200 superframes in our example), as shown on Figure 3.17 (a) and (b). It can be noticed here that below a covariance of about 2m² the location error is systematically lower than 2m, hence justifying the presumed reliability of elected virtual anchors based on the estimation covariance indicator.

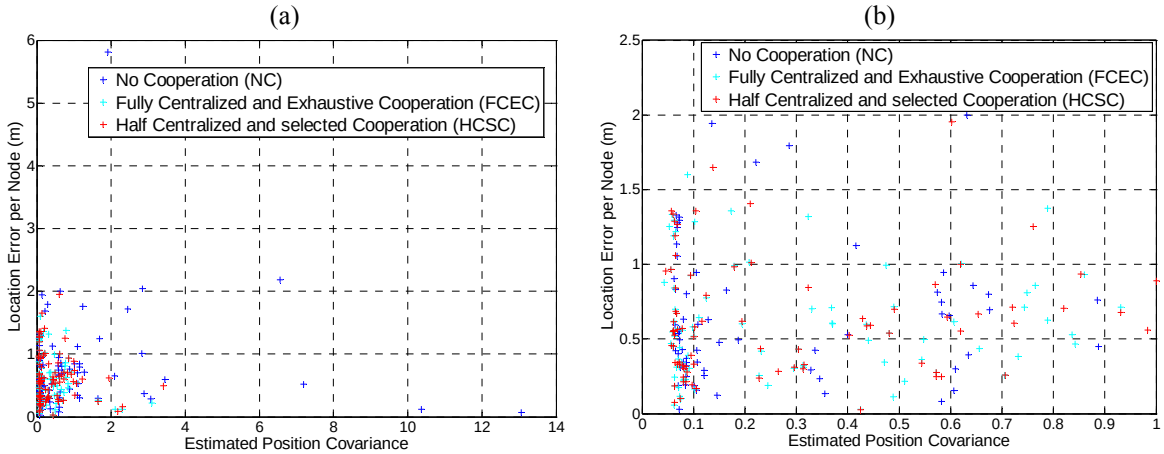


Figure 3.17: Instantaneous location error per node as a function of the estimated location covariance (in m²) (a), at *t*=60s, and zoom on the smallest values (b).

Moreover, as our prioritized MAC protocol is partly based on mobility, it is worth showing the distribution of the estimated location covariance (i.e. the locally perceived location uncertainty) as a function of node speed (See Figure 3.18 (a)). We can note that for the NC scheme, the lowest velocities have a higher covariance, mostly resulting form a lack of observations (i.e. a low number of available range measurements). Concerning the actual location error as a function of the mobile speed (See Figure 3.18 (b)), as expected, the distribution seems to be almost uniform. The high-speed mobiles are favoured through prioritized access and perform a sufficient number of ranging transactions to handle their position variability, whereas the low-speed mobiles deliberately deteriorate their ranging refreshment rate, with no significant degradation of their estimated location.

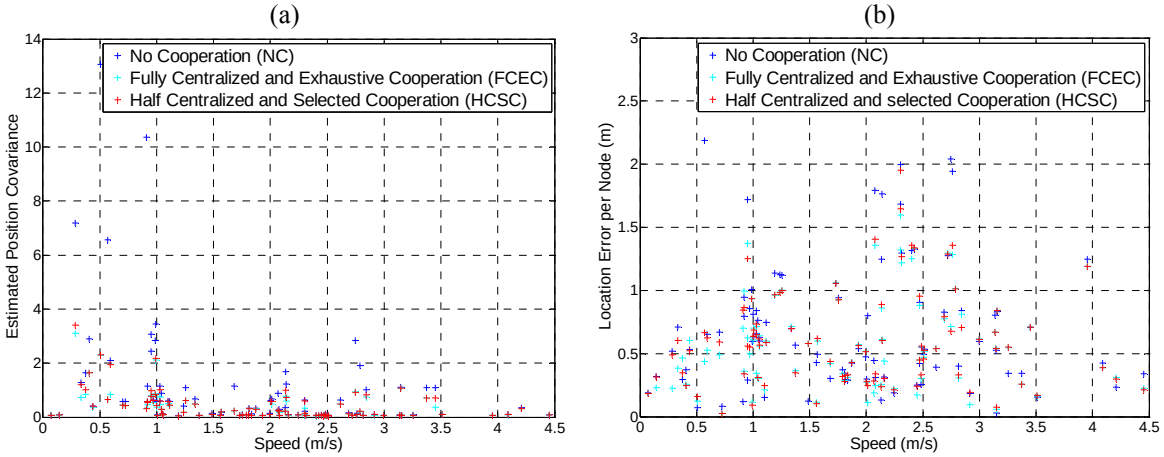


Figure 3.18: Instantaneous estimated location covariance (in m²) per node (a) and actual location error (b), at *t*=60s, as a function of experience 2D speed.

As for decentralized cooperation schemes, performance is also characterized here in terms of global network location error (e.g. the average location error of all the mobile devices at a given superframe). In order to compare the fully decentralized scheme with the NC scheme under the same conditions in the response period (collisions, priority setting and refreshment rate), we simulate a common scenario in terms of mobility and measurement noises for the two following Extended Kalman Filters:

- FDSC filter: We consider the whole observation vector, i.e. the successful ranging with respect to each neighbouring anchors or virtual anchors.
- NC filter: We consider only the observations with respect to the neighbouring anchors.

Figure 3.19 compares the global network location error for FDSC as a function of the elapsed time, with that of NC, showing significant gain through cooperation even within decentralized schemes. In this example, the scaling factor for the probability function of virtual anchor self-election is set to c_s=0.03.

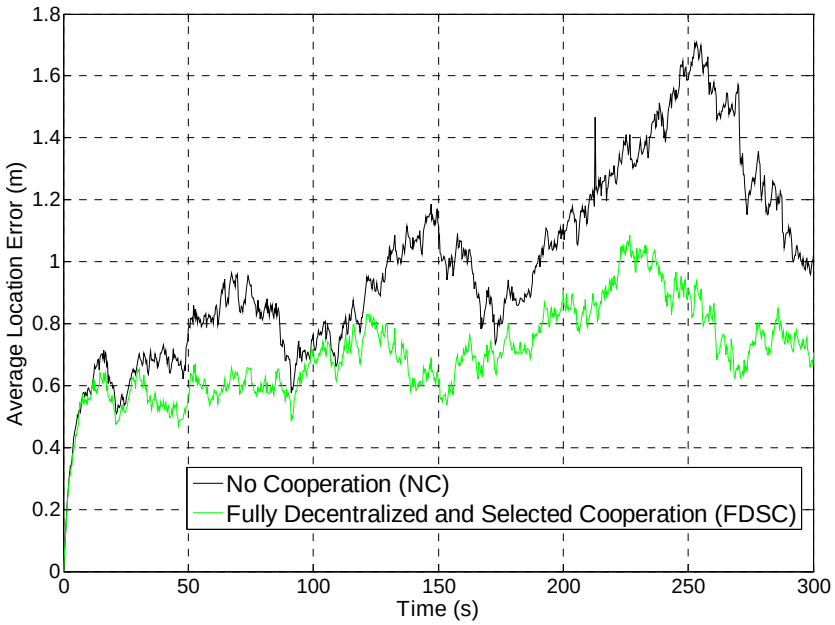


Figure 3.19: Example of global average location error (over all the mobile nodes) for FDSC and NC schemes, as a function of the elapsed time with perfect initialization.

Figure 3.20 displays the CDF of the location error for FDSC and NC at a given time t=60s (i.e. after 200 superframes). In the example, 95% of the mobile devices achieve a location error lower than 1.35m for HCSC (resp. 1.85m for NC).

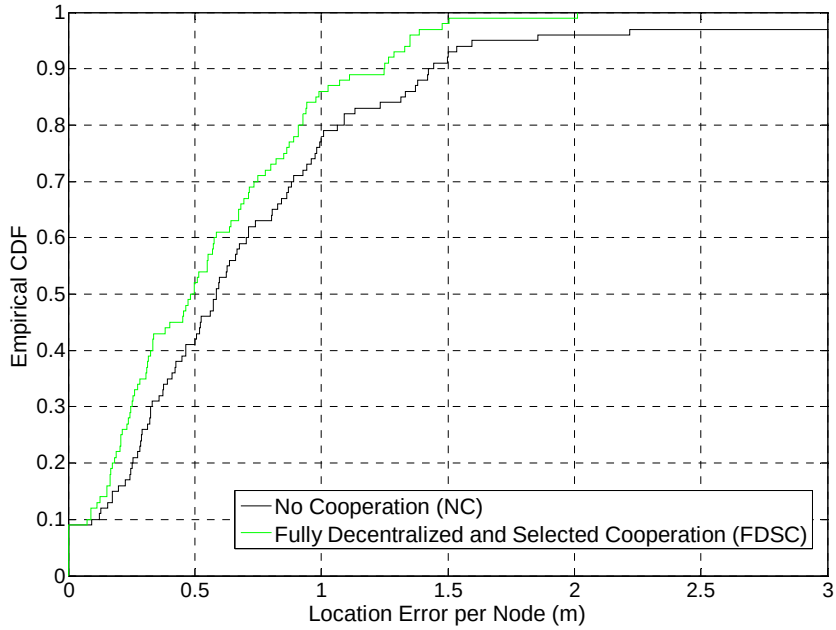


Figure 3.20: CDF of location errors after 200 Superframes for decentralized cooperative and non-cooperative schemes.

In the distributed FDSC scheme, devices decide if they have access or not to the cooperative slots of the request period. Setting a fixed number of cooperative slots in the request and acknowledgement period, we remind that the aim of the probability function P_{VA}^{k} (through c_s) is to select some mobile devices with the lowest position covariance and to allow them to transmit their ranging packet as virtual anchors (See Figure 3.21). Under the fixed number of cooperative slots, a possible enhancement could be to define a dynamic or adaptive scaling factor c_s. Actually, at different times during the simulation, a constant a priori scaling factor proved not to be always relevant and adapted to the context. On the one hand, the number of elected virtual anchors at a given superframe is most probably too high. This implies collisions during the request and the acknowledgement periods and disables the benefits from cooperation. On the other hand, the number of elected virtual anchors at a given superframe can be too low. Thus, many cooperative slots can be unused.

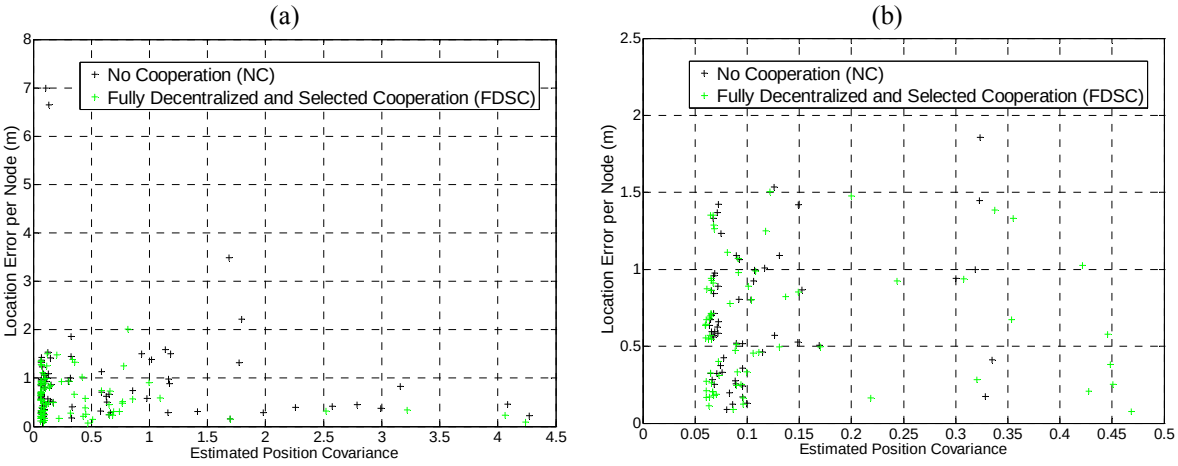


Figure 3.21: Instantaneous location error per node as a function of the estimated location covariance (in m²) (a), at *t*=60s, and zoom on the smallest values (b).

Finally, we compare directly the performances of the previous centralized and decentralized protocol schemes, averaged over random simulation trials and we show the minimal and maximal location errors with so-called "perfect" (Figure 3.22) and random initialization (Figure 3.23 (a) and (b)). For the random initialization, the starting location error is thus around 20m for the 40 m*40m area of interest. In this case, after less than 25 seconds, the global network location error falls below 1m with our access scheme, what represents a latency gain of at least 50% in comparison with non-cooperative schemes.
In both cases, one can note that cooperative techniques also improve performance in comparison with non-cooperative tracking, in terms of both mean error and error dispersion (i.e. the difference between min and max observed errors over random simulation trials). Under random initialization for instance (i.e. starting from scratch), the average error floor achieved after convergence is reduced by more than 40% with cooperative access schemes (i.e. from about 1.25m with non-cooperative tracking down to about 0.7m). The 3 cooperative schemes enjoy very similar results, but at the price of different energy consumptions and different system requirements (e.g. the fully decentralized scheme is much lighter than the two other schemes). Finally, note that cooperation mostly improves estimation performances for the nodes which are poorly estimated under non-cooperative schemes (from infrequent/erroneous ranging measurements, and/or poor GDOP…).

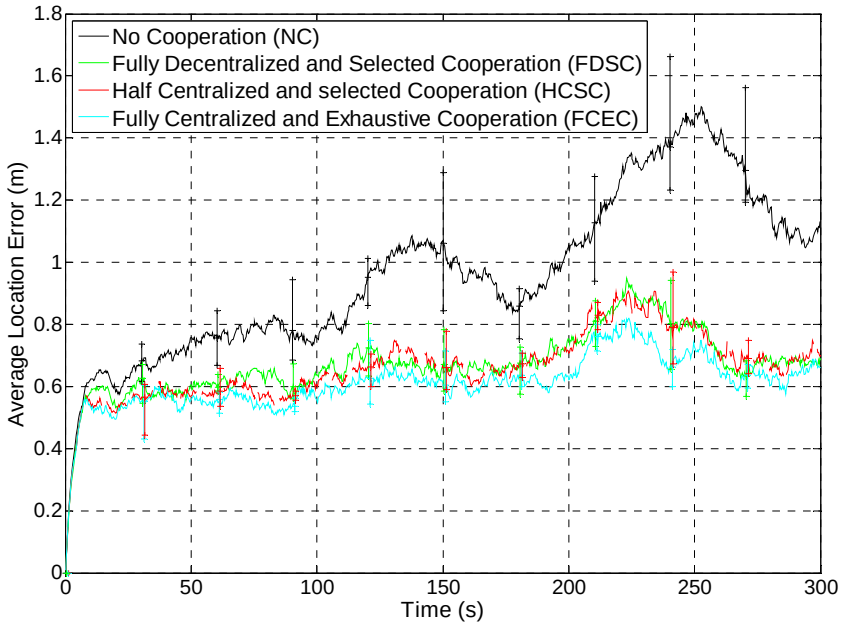


Figure 3.22: Average location error (over all the mobile nodes and different simulation trials), as a function of the elapsed time and with perfect initialization.

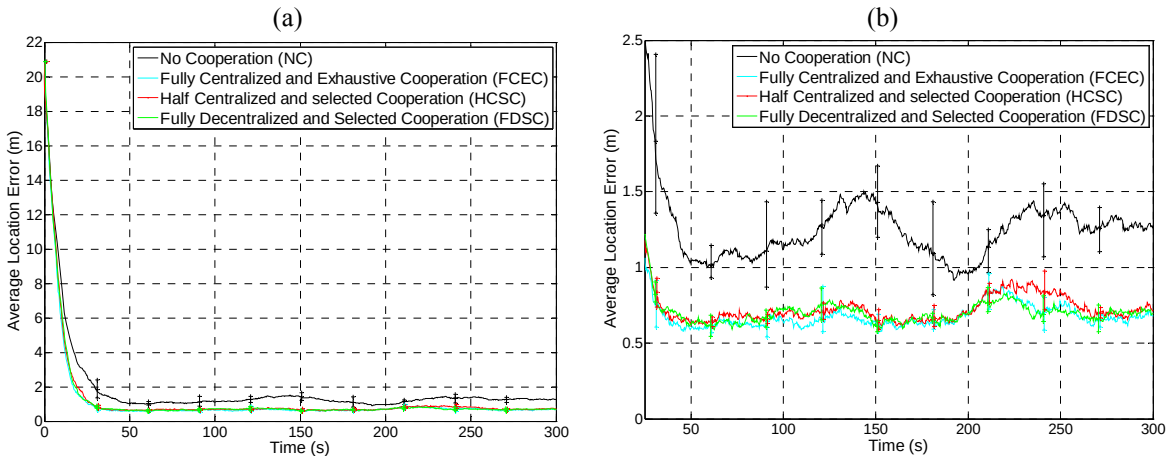


Figure 3.23: Average location error (over all the mobile nodes and different simulation trials), as a function of the elapsed time with random initialization (a) and zoom on the steady-state regime after convergence is achieved (b).

3.2.5 Conclusion and Perspectives

Prioritized medium access schemes enabling decentralized filtering under partial connectivity and energetic constraints have been successfully extended into cooperative tracking scenarios. These developments are mostly intended with short-range cooperative links, e.g. like in the T1-A scenario (but with no consideration with respect to fusion with extra cellular means). However, some of the shown cooperative concepts (including both resource allocation and filtering) could be adapted to cope with HDF in future heterogeneous networks.

In the shown canonical example, the proposed access schemes, coupled with a specific filtering formulation, make the steady-state average location error (i.e. after convergence) decrease by more than 40%, while initial latency (i.e. the time to achieve a target average location precision of e.g. 1m, starting from scratch with random initial guess) can be reduced by about 50% in comparison with non-cooperative tracking (even with prioritized access).

Besides the obtained results, several axes of enhancement have been identified. First of all, centralized cooperation is likely to provide more spatial diversity (through Round Robin allocation), whereas the decentralized scheme tends to use always the same nodes as virtual anchors after a while. Another issue with the chosen selective cooperative schemes (HCSC and FDSC) is that the decision is made so far independently of the actual mobiles' neighbourhood needs but it is mostly based on the presumed cooperation gains offered locally (even if neighbours do not need so). One related problem is that cooperative ranging requests are initiated by virtual anchors. However, mitigating the previous remark, it is worth mentioning that the number of cooperative slots N_c has been kept constant in all the study, most probably at a too high level for the decentralized scheme, so that the gain in latency (number of superframes before the convergence is achieved) with random initialization is not as spectacular as expected. A more judicious trade-off still might be found as regards to this N_c value. Finally, the virtual anchor election could be also based on the success rate of past cooperation attempts. This could allow a better turn-over of elected cooperative devices. Moreover another possible enhancement could consist in considering jointly the local priority setting (i.e. used for ranging responses) and the need for active neighbouring virtual anchors. Finally, for the definition of the high priority level, which depends only on speed in this study, the estimation covariance could be an interesting parameter as well. Some of these identified improvements should be investigated in the frame WHERE2 WP2.

As another perspective, other Game-theoretical approaches [GKAug08] [GKJuly08] (e.g. based on the Shapley value and weighted-graph games…) or complementary studies addressing smart links selection (e.g. [MR06], [K06]), which have been mostly considered for non-cooperative scenarios so far, could be combined and extended so as to provide even more efficient support of communication means and limit harmful reciprocal impact with respect to location. In this context, mobility and network heterogeneity might be the crucial issue. This point should be addressed as well in the frame of WHERE2 WP2.

4. Conclusion and Future Work

4.1 Conclusion

In general, this deliverable presented the final results from WHERE WP2 Task 2.2 of the WHERE project. The presented results about positioning accuracy and communication overhead should serve as an input to WHERE WP3. The major contributions were in two folds: novel cooperative positioning schemes for various communication architectures and performance analysis from the communication aspects. The focus of this deliverable was on four WHERE scenarios, in particular the small-scale indoor scenarios T1.A and T1.B, the mid-scale indoor scenario T2.A, and the large scale outdoor scenario T3. It was shown that the proposed cooperative positioning schemes such as iterative cooperative positioning algorithms, generalized belief propagation (GBP), the mean field algorithm, and multidimensional scaling can significantly improve the accuracy of positioning information in the small-scale scenarios, respectively, with the pay of communication overhead. The benefit of cooperation is verified using IR-UWB hardware platforms. Trade-off between positioning accuracy and communication overhead is carefully investigated for both the mid-scale (WiFi) and large-scale scenarios (cellular). Medium access schemes for cooperative positioning are proposed to reduce consumed energy and time resources and relax coordination needs, but maintain the overall accuracy of location information through cooperation.

Specifically, Section 2 presented novel cooperative positioning schemes for various wireless environments. Specifically, Section 2.1 presents multi-hop cooperative positioning schemes for ad hoc networks. A distributed iterative multilateration approach has been introduced. It requires a relatively high degree of connectivity in order to localize all nodes in a small number of iterations. The major drawback of this approach is error propagation and accumulation, resulting from measurement errors, and from the fact that erroneous virtual anchors are used as references. In order to avoid error propagation, a metric for virtual anchor uncertainty has to be modeled. Also anchor selection plays a great role in improving accuracy. Using reference nodes that are well separated helps to reduce the positioning error in least square based calculation.

Section 2.2 presented the GBP localization technique for scenario T1.B. This section proposed four methods: GBP based on Kikuchi approximation (GBP-K), GBP based on junction-tree method (GBP-JT), nonparametric GBP-JT (NGBP-JT) and NBP based on spanning trees (NBP-ST). It was shown that the last one (NBP-ST) is currently the unique method which is computationally feasible in large-scale ad-hoc/sensor networks. In addition, real database was used in order to obtain more realistic model for indoor scenario. The obtained models justify importance of all probabilistic methods since they are capable to handle non-Gaussian uncertainties.

As an alternative to the BP algorithm, Section 2.3 presented cooperative positioning using the mean field algorithm. This section used variational methods and mean field theory to develop an algorithm for sensor self-localization in cooperative wireless networks.

In Section 2.4, cooperative positioning and tracking algorithms were analyzed under realistic communications constraints for scenario T2.A. It was shown that the introduction of realistic communications constraints resulted in an added delay, which had a significant effect on the positioning performance, especially for the cooperative algorithms. It was found that the static solution and the extended Kalman filter algorithms were similarly affected by the realistic communications constraints. Further, it was observed that increasing the number of cooperating mobile stations had a positive impact on the positioning performance, as expected due to added cooperation possibilities. However, this was only until a tipping point was reached and the performance became worse with additional cooperating mobile stations. Nevertheless, in most cases the cooperative approach strongly outperformed the conventional (non-cooperative) approach.

Section 2.5 presented two cooperative localization algorithms in the scenario T3 (cellular radio systems). The basic idea of proposed algorithms is utilizing located mobiles to serve as anchors. The proposed algorithms can significantly reduce the training and signaling overhead paid for localization in cellular network, without significantly degradation in accuracy.

Section 2.6 described localization with multidimensional scaling (LMDS) which is a simple algebraic algorithm for cooperative localization. The main advantages of this algorithm are simplicity, exploiting of redundant information, and ability to fuse different radio parameters. Indeed, the step of ranges collection (symmetric matrix D) can be done using different radio parameters (RSSI, TOA, or TDOA) leading to an estimation of ranges between different pairs of sensors. Nevertheless, LMDS estimates improve as

ranging improves. Moreover, there may be scenarios with some missed ranges which lead to a non-complete matrix D. This may occurs when no measurement is detected between two sensors. In these cases, the matrix may be completed using some geometric relations (this solution may be heavy) or some matrix completion techniques.

Section 2.7 practically illustrated the benefits of cooperative positioning with respect to positioning precision. This gain is verified in harmful environments that would adversely lead to strongly biased measurements. In comparison with non-cooperative approaches, the gain is even more noticeable in case of generalized biased measurements. Then, this section investigated the importance of the relative geometrical configuration of both virtual anchors and other mobiles, which might locally alter cooperation gains for certain mobile nodes. This remark tends to justify the general efforts made in WHERE WP2 Task 2.2 aiming at more advanced links selection/discarding schemes identifying the most relevant cooperative measurements and neighbours (see also [D22] and [DM09]). Finally, it has been shown that the gain on the location error after removing systematic biases could be spectacular, emphasizing the weakness of simple LS approaches in case of NLOS situations and the needs for more advanced bias mitigating techniques in tracking or positioning, as investigated in WHERE WP2 Task 2.1 [D23] and WHERE WP4 Task 4.2 and Task 4.3.

Section 3 presented performance of cooperative positioning techniques from a communication-oriented perspective. Specifically, Section 3.1 described the realistic communication constraints for the conventional and cooperative localization algorithms considered in Section 2.4 and evaluate network related performance metrics of the communication required for localization. It was shown that the introduction of realistic communications constraints resulted in an added delay, which is shown to have a significant effect on the positioning performance in Section 2.4, especially for the cooperative algorithms. In Section 2.4, we also observed that increasing the number of cooperating MSs had a positive impact on the positioning performance, as expected due to added cooperation possibilities. However, this was only until a tipping point was reached and the performance became worse with additional cooperating MSs. This tipping point is likely a result of the communication overhead becoming large, which in turn leads to increased delays.

Finally in Section 3.2, prioritized and decentralized medium access schemes were extended into the cooperative tracking context, showing promising location errors and convergence properties and benefiting from relaxed energy consumption and system requirements (e.g. in terms of coordination and synchronization). In comparison with classical TDMA-based schemes, the proposed uncoordinated and decentralized prioritized solutions appear to be compliant with some critical non-cooperative tracking requirements in large-scale networks, under changing connectivity conditions. More particularly, these protocols still tend to favour high-speed targets as regards to the success rate and precision of the ranging measurements issued at anchor nodes in the nominal non-coopertaive mode, while providing opportunistically further virtual anchors to the most demanding mobiles (in terms of expected inaccuracy).

4.2 Future Work

Besides others, particular the following issues will be part of future work: for Section 2.2, there remain many open directions for the future work. The most important is the generalizing NGBP-JT method for large-scale ad-hoc/sensor networks using some efficient method for formation of junction tree cliques within the network. Moreover, including RSS/TOA indoor data, in some of the described algorithms, will provide more precise conclusions about the performance. Finally, real-time target tracking using these methods could be an interesting direction.

For Section 2.3, future work includes verifying the localization algorithm by simulations and against real measurement data. Generalizing the algorithm to three dimensions and including more data (e.g. AoA) could improve the accuracy of the position estimates. Furthermore, tracking of moving mobile nodes is an interesting extension to this work.

For Section 2.6, future work will validate even further the actual gain from cooperation in more complex environments, under mobility and network heterogeneity.

For Section 3.2, for future work, other Game-theoretical approaches [GKAug08] [GKJuly08] (e.g. based on the Shapley value and weighted-graph games…) or complementary studies addressing smart links selection (e.g. [MR06], [K06]), which have been mostly considered for non-cooperative scenarios so far, could be combined and extended so as to provide even more efficient support of communication means

and limit harmful reciprocal impact with respect to location. In this context, mobility and network heterogeneity might be crucial issues.

5. References

[AMG+02] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on particle filters for online nonlinear/non-Gaussian bayesian tracking", IEEE Transactions on Signal Processing, vol. 50, issue 2, pp. 174-188, February 2002.

[APAL06] N.Aslindi, K.Pahlavan, B.Alavi, X.Li: "A novel cooperative localization algorithm for indoor sensor networks", in IEEE International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC '06, Helsinki, Finland, pp.1-6, 2006.

[Bishop2006] C. M. Bishop, "Pattern Recognition and Machine Learning", ISBN-10: 0-387-31073-8, ISBN-13: 978-0387-31073-2, Springer Science, 2006.

[BM06] D.A.Bader, K. Madduri, "Designing Multithreaded Algorithms for Breadth-First Search and st-connectivity on the Cray MTA-2", in IEEE Proc. of Parallel Processing - ICPP, pp. 523-530, August 2006.

[CLTZ07] Q. Cui, J. Liu, X. Tao, and P. Zhang, "A novel location model for 4G mobile communication networks," 2007.

[CS09] F. W. C. Chan and H. C. So, "Accurate Distributed Range-Based Positioning Algorithm for Wireless Sensor Networks," IEEE Transactions on Signal Processing, vol. 57, no. 10, pp. 4100–4105, October 2009.

[CSJ+07] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, and H. Hartenstein, "Overhaul of IEEE 802.11 modeling and simulation in ns-2," Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems, 2007.

[D11] ICT-217033 WHERE Deliverable D1.1, "Definition of the WHERE framework and scenarios", March 2008.

[D22] ICT-217033 WHERE Deliverable D2.2, "Cooperative Positioning - Intermediate Report", March 2009.

[D23] ICT-217033 WHERE Deliverable D2.3, "Hybrid Localisation Techniques", April 2010.

[D41] ICT-217033 WHERE Deliverable D4.1, WHERE Project, "Measurements of Location-Dependent Channel Features", October 2008.

[D45] ICT-217033 WHERE Deliverable D4.5, "Modelling of the Channel and its Variability (Final Report)," April 2010.

[D46] ICT-217033 WHERE Deliverable D4.6, "Estimation of Location-Dependent Channel Information (Final Report)", April 2010.

[D53] ICT-217033 WHERE Deliverable D5.3, "Evaluation Campaigns Report 2", Feb. 2010.

[DKZ+03] P.M. Djuric, J.H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M.F. Bugallo, J. Miguez, "Particle filtering", IEEE Signal Processing Magazine, vol. 20, issue 5, pp. 19-38, September 2003.

[DM09] B. Denis, M. Maman, L. Ouvry, "On the Scheduling of Ranging and Distributed Positioning Updates in Cooperative IR-UWB Networks", in Proc. IEEE ICUWB'09, Vancouver, Sept. 2009.

[FF07] S. Frattasi and J. Figueiras, "Ad-coop positioning system: using an embedded Kalman filter data fusion," CRC Press, 2007.

[FIGU08] J. Figueiras, "Accuracy enhancements for positioning of mobile devices in wireless communication networks," PhD dissertation, Aalborg University, Jan. 2008.

[FRA07] S. Frattasi, "Link Layer Techniques Enabling Cooperation in Fourth Generation Wireless Networks," Ph.D. dissertation, Aalborg University, Aalborg, Denmark, September 2007.

[FRA07] S. Frattasi, "Link layer techniques enabling cooperation in fourth generation wireless networks," PhD dissertation, Aalborg University, Sept. 2007.

[GC09] I. Guvenc and C.C. Chong, "A survey on TOA based wireless localization and NLOS mitigation techniques", IEEE Communications Surveys and Tutorials, vol. 11, no. 3, pp. 107.124, 2009.

[GG05] F. Gustafsson and F. Gunnarsson, "Mobile Positioning Using Wireless Networks," IEEE Signal Processing Magazine, vol. 22, no. 4, pp. 41–53, July 2005.

[GKAug08] F. Ghassemi and V. Krishnamurthy, "Decentralized Node Selection for Localization in Wireless Unattended Ground Sensor Networks", in Proc. IEEE SENSORCOMM'08, Aug. 2008.

[GKJuly08] F. Ghassemi and V. Krishnamurthy, "A Cooperative Game-Theoretic Measurement Allocation Algorithm for Localization in Unattended Ground Sensor Networks", in Proc. IEEE ICIF'08, July 2008.

[IEEE80211] "Wireless lan medium access control (mac) and physical layer (phy) specifications," IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999), pp. C1–1184, 12 2007.

[IEEEstd06] "IEEE Standard for Information technology- Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)", IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003), pp. 1-305, 2006.

[Ihler05] A.T. Ihler, "Inference in Sensor Networks: Graphical Models and Particle Methods", Thesis, MIT, Department of Electrical Engineering and Computer Science, June 2005.

[JW02] M.I. Jordan and Y. Weiss, "Graphical model: probabilistic inference", The Handbook of Brain Theory and Neural Networks, 2nd edition, Cambridge, MA, MIT Press, 2002.

[K06] L. M. Kaplan, "Local Node Selection for Localization in a Distributed Sensor Network", IEEE Trans. on Aerospace and Electronic Systems", Jan. 2006.

[KAY93] S. M. Kay, Fundamentals of Statistical Signal Processing: Estimation Theory. Prentice Hall, 1993.

[KLH+06] T. Kim, H. Lim, and J. Hou, "Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks," in Proceedings of the 12th annual international conference on Mobile computing and networking. ACM, 2006, p. 377.

[LAA09] M. Laaraiedh, S. Avrillon, and B. Uguen. "Enhancing positioning accuracy through RSS based ranging and weighted least square approximation". Proceedings of POCA conference, Antwerp, Belgium, May, 2009.

[LAR04] E. G. Larsson, "Cramer-Rao bound analysis of distributed positioning in sensor networks," in IEEE Signal Processing Letts., 2004.

[MDO08] M. Maman, B. Denis, L. Ouvry, "An Intuitive Prioritised Medium Access Scheme for Tracking Applications in UWB LDR-LT Networks", in Proc. IEEE PIMRC'08, Cannes, Sept. 2008.

[Mensing10] C. Mensing et al. "Performance Assessment of Cooperative Positioning Techniques", in Proc. ICT Future Networks and Mobile Summit, Florence, June 2010.

[Minka2005] T. Minka, "Divergence Measures and Message Passing", Microsoft Research Technical Report MSR-TR-2005-173, December 2005.

[MK07] J. M. Mooij and H. J. Kappen, "Sufficient conditions for convergence of the sum-product algorithm", IEEE Transaction on Information Theory, vol. 53, issue 12, pp. 4422-4437, December 2007.

[MR06] A. I. Mourikis and S. I. Roumeliotis, "Optimal Sensor Scheduling for Resource-Constrained Localization of Mobile Robot Formations", IEEE Trans. on Robotics, Oct. 2006.

[MRM+07] C. L. F. Mayorga, F. della Rosa, S. A. Wardana, G. Simone, M. C. N. Raynal, J. Figueiras, and S. Frattasi, "Cooperative Positioning Techniques for Mobile Localization in 4G Cellular Networks," Proceedings of the IEEE International Conference on Pervasive Services, July 2007.

[MSD+09] C. Mensing, S. Sand, A. Dammann, and W. Utschick, .Interference-Aware Location estimation in Cellular OFDM Communications Systems., in Proceedings of the IEEE International Conference on Communications (ICC), Dresden, Germany, June 2009.

[NN01] D. Niculescu, B. Nath, "Ad hoc positioning system (APS)", in GLOBECOM, pp.2926-2931, 2001.

[PAK+05] N. Patwari, J.N. Ash, S. Kyperountas, and A.O. Hero III, "Locating the nodes: cooperative localization in wireless sensor networks", IEEE Signal Processing Mag., vol. 22, no. 4, pp. 54.69, Jul. 2005.

[Pearl88] J. Pearl, "Probabilistic Reasoning in Intelligent Systems. Networks of plausible inference", Morgan Kaufmann, 1988.

[PIC08] F.L. Piccolo, "A New Cooperative Localization Method for UMTS Cellular Networks", in GLOBECOM, New Orleans, LA, USA, December 2008.

[PP07] T. Perälä and R. Piché, "Robust Extended Kalman Filtering in Hybrid Positioning Applications," Proceedings of the Workshop on Positioning, Navigation and Communication (WPNC), March 2007.

[PULSII08] G. Abreu, G. Destino, D. Macagnano, C. Lima, S. Khan, B. Denis, M. Maman, "Novel Solutions for MAC Scheme", Deliverable D3a-6.2 of the PULSERS II project (ICT-IP n°027142).

[QUA81] A. Quazi, "An overview on the time delay estimation in active and passive systems for target localization", IEEE Tran. Acoustics, Speech and Signal Processing, vol. 29, no. 3, Jun. 1981.

[Rap96] T.S. Rappaport, "Wireless communications – principles and practice", Prentice-Hall, Inc, New Jersey, 1996.

[RJ07] V. Rodriguez, F. Jondral, “Simple, adaptively-prioritised, spatially-reusable medium access control through the Dutch auction: Qualitative analysis, issues, challenges”, Nov. 2007.

[SCG+05] G.Sun, J. Chen, W. Guo, and K. J. R. Liu, “Signal processing techniques in network-aided positioning”, IEEE Signal Processing Mag., vol. 22, no. 4, pp. 12.23, Jul. 2005.

[SHA03] Shang, Ruml, Zhang, and Fromherz. “Localization from mere connectivity”. In MobiHoc, 2003.

[SMP03] S. Slijepcevic, S. Megerian, M. Potkonjak, “Characterization of location error in wireless sensor networks: analysis and applications”, IPSN ’03, Palo Alto, CA, USA, pp.593-608, 2003.

[SZ09a] V. Savic and S. Zazo, “Sensor localization using generalized belief propagation in network with loops”, in Proc. of the 17th European Signal Processing Conference - EUSIPCO, pp. 75-79, August 2009.

[SZ09b] V. Savic and S. Zazo, “Sensor localization using nonparametric generalized belief propagation in network with loops”, in IEEE Proc. of the 12th IEEE International Conference on Information Fusion, pp. 1966-1973, July 2009.

[Weiss00] Y. Weiss, “Correctness of local probability propagation in graphical models with loops”, in Neural Computation, vol. 12, issue 1, pp. 1-41, January 2000.

[WF01] Y. Weiss and W.T. Freeman, “Correctness of belief propagation in Gaussian graphical models of arbitrary topology”, in Neural Computation, vol. 13, issue 10, pp. 2173-2200, October 2001.

[WJW01] M.J. Wainwright, T. Jaakkola, and A.S. Willsky, “Tree-based reparameterization for approximate inference on loopy graphs”, in Advances in Neural Information Processing Systems, pp. 1-8, 2001.

[WJW03] M.J. Wainwright, T. Jaakkola, and A.S. Willsky, “Tree-based reparameterization framework for analysis of sum-product and related algorithms”, IEEE transactions on Information theory, Vol. 49, No. 5, pp. 1120-1146, May 2003.

[WLW09] H. Wymeersch, J. Lien, and M. Z. Win, “Cooperative Localization in Wireless Networks,” Proceedings of the IEEE, February 2009.

[WYG06] Y.Wei, Z.Yu, Y.Guan, “COTA: A robust multi-hop localization scheme in wireless sensor networks”, in Proc. of IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS ´06).

[YFW03] J.S. Yedidia, W.T. Freeman, and Y. Weiss, “Understanding belief propagation and its generalizations”, Exploring artificial intelligence in the new millennium, ACM, pp. 239-269, January 2003.

[YFW05] J.S. Yedidia, W.T. Freeman, and Y. Weiss, “Constructing free energy approximations and generalized belief propagation algorithms”, IEEE Transaction on Information Theory, vol. 51, issue 7, pp. 2282-2312, July 2005.

[ZHL2006] Y. Zhang, Q. Huang, J. Liu, “Sequential localization algoritm for active sensor network deployment”, in Proc. of 20th International Conference on Advanced Information Networking and Applications (AINA ’06), pp. 171-178, 2006.